



CENTRE DE RENNES

IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105

78153 Le Chesnay Cedex
France

Tél. (3) 954 90 20

Rapports Techniques

N° 66

MADMACS
MANUEL D'UTILISATION

Eric GAUTRIN

Mars 1986

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

MADMACS

Manuel d'utilisation

Janvier 1986

ERIC GAUTRIN

IRISA.API
Campus de Beaulieu
35042 RENNES Cedex
France

Tel : 99 36 20 00 poste 460
Adresse Multics : Gautrin.API -at CICB

Résumé : Madmacs est un éditeur graphique interactif conçu pour le dessin de circuits VLSI hiérarchiques. Comme la plupart des éditeurs, il inclut toutes les fonctions d'édition nécessaires à la conception de circuits complexes. De plus, Madmacs offre à l'utilisateur une interface puissante qui facilite et accélère l'édition de dessins réguliers et structurés. Cette technique, similaire au concept de macro édition utilisée dans l'éditeur de texte EMACS, est appelée PIA : Procédure d'Implantation Automatique. Elle permet de redessiner automatiquement le circuit suite à de légères modifications. Dans cet article, nous décrivons les caractéristiques générales de Madmacs ainsi que nos premiers résultats issus d'une réalisation expérimentale.

Abstract : MADMACS is an interactive graphics editor for hierarchical VLSI layouts. Like most editors, it includes all editing facilities needed for complex circuit design. In addition, Madmacs provides a powerful user interface to facilitate and speedup regular and structured layout editing. This technique, similar to the macro editing concept used in the EMACS text editor is called ALP : Automated Layout Procedure. It permits the easy update of a design when only some slight modifications are required. In this paper, we describe the general features of Madmacs and our first experiences with a preliminary implementation.

MADMACS : MANUEL D'UTILISATION

E ric GAUTRIN

NOTE TECHNIQUE 8601

JANVIER 1986



Sommaire

p. 04	INTRODUCTION
p. 05	1 Présentation générale
p. 05	1.1 L'appel de MADMACS
p. 05	1.2 L'interface utilisateur
p. 07	1.3 Le curseur
p. 08	2 Déplacements simples du curseur
p. 09	3 Les objets manipulés
p. 09	3.1 Cellule-Figure-Rectangle
p. 10	3.2 Création d'une cellule
p. 10	3.2.1 La marque courante
p. 11	3.2.2 Création d'un rectangle
p. 11	3.2.3 Création d'une figure
p. 12	3.2.4 La cellule pointée
p. 12	3.3 La manipulation de la hiérarchie
p. 12	3.3.1 Les déplacements dans la hiérarchie
p. 13	3.3.2 Les informations graphiques sur les figures
p. 13	3.3.3 La modification de la hiérarchie
p. 14	3.4 Destruction d'une cellule
p. 15	3.5 Modification du dessin
p. 15	3.5.1 Modification de la taille d'un rectangle
p. 16	3.5.2 Déplacement de cellule
p. 16	3.5.3 Orientation de cellule
p. 16	3.5.4 Reproduction d'une cellule
p. 17	3.5.5 Changer la couleur d'un rectangle
p. 18	4 Les déplacements relatifs du curseur
p. 19	4.1 Déplacement inter-cellulaire
p. 19	4.2 Déplacement intra-cellulaire
p. 20	4.3 Alignement par rapport à une autre cellule
p. 20	4.4 Déplacement à un connecteur
p. 21	5 Les marques
p. 21	5.1 La marque courante
p. 22	5.2 Les marques nommées
p. 23	6 Le nommage des objets
p. 23	6.1 Nommage d'une figure
p. 23	6.2 Nommage d'un rectangle

p. 24	7 Les commandes de visualisation
p. 24	7.1 Recadrage du dessin
p. 24	7.2 Recadrage du curseur
p. 25	7.3 Déplacement de l'écran par demi-fenêtres
p. 25	7.4 Plan de masse
p. 25	7.5 Restriction de l'affichage
p. 26	8 Les tampons
p. 26	8.1 Ouverture d'un buffer
p. 26	8.2 Gestion des buffers
p. 27	9 Les scripts
p. 28	10 L'environnement système
p. 28	11 Outils de vérification NMOS
p. 29	12 Les macros
p. 34	13 Les Procédures d'Implantation Automatique
p. 34	13.1 Construction d'un PIA
p. 35	13.2 Exécution et exemple
p. 37	14 Méthodologie de conception
p. 38	CONCLUSION
p. 39	ANNEXE 1 : Commandes de MADMACS par types
p. 46	ANNEXE 2 : Commandes MADMACS par ordre lexicographique
p. 49	ANNEXE 3 : Erreurs détectées par le vérificateur de dessins.
p. 51	ANNEXE 4 : Différentes constantes du système

INTRODUCTION

Le nombre de transistors intégrés sur une pastille de silicium double tous les deux ans. Les concepteurs de circuits doivent posséder des outils d'aide performants à chaque étape de la conception (outils de CAO) pour faire face à la complexité croissante des circuits. L'outil développé au sein de l'équipe API-IRISA est dans sa version actuelle un éditeur de dessins orienté conception de circuits VLSI. Malgré une gamme bien fournie d'éditeurs graphiques, MADMACS présente des nouveautés et des points forts par rapport à ses concurrents.

Un bon outil de CAO est un outil interactif répondant par des commandes simples. MADMACS répond parfaitement à ces sollicitations. Il intègre de plus la notion de hiérarchie, permettant de dériver un circuit, depuis son plan de masse jusqu'au dessin de masques complet. MADMACS propose des commandes simples pour la construction de structures régulières [Mead 1980]. MADMACS présente une originalité pour les déplacements du curseur: outre des déplacements basés sur le paramètre λ , il permet des déplacements relatifs aux dimensions des objets manipulés et aux distances entre ces objets. Notre éditeur intègre, de plus, deux mécanismes: le premier bien connu dans les éditeurs de texte permet la construction et la réexécution de macrocommandes (macros), le deuxième plus original appelé Procédures d'Implantation Automatique (PIA) permet de paramétrer une construction.

La mise en oeuvre de MADMACS est un travail d'équipe, outre moi-même trois autres personnes participent (ou ont participées) à la réalisation de ce nouveau système: Patrice Quinton d'une part pour le vérificateur de règles de dessin en technologie NMOS, Philippe Duvivier pour l'extracteur de circuits électriques en technologie NMOS et enfin Patrice Frison dont la participation va des deux outils cités ci-dessus jusqu'à MADMACS.

Nous présentons dans ce document les différents concepts de MADMACS en détaillant la majeure partie des commandes du système. Cette note permettra au lecteur d'estimer la puissance de MADMACS et de se familiariser aux concepts intégrés dans le système. Il est de plus fortement conseillé d'utiliser en parallèle MADMACS, et de réaliser les manipulations proposées tout au long de ce manuel.

1 Présentation générale

1.1 L'appel de MADMACS

MADMACS est développé actuellement sur le système MULTICS, et en exploitation sur HP9000 et SM90. Il est en cours de transport sur différentes machines : VAX 780, RIDGE, SUN....

Avant d'appeler MADMACS, vérifiez que vous avez bien les accès sur le programme madmacs, sur les fichiers de commandes (primaire.com, ctrlx.com, ctrlz.com, escape.com), et sur le fichier aide-mémoire (docab). De plus vérifiez que le type de terminal employé est reconnaissable par MADMACS. Pour plus de renseignements, consultez l'administrateur du logiciel.

Pour lancer MADMACS, tapez la commande :

ec >udd>API>vlsi>madmacs sur Multics.

madmacs sur HP9000 ou SM90.

1.2 L'interface utilisateur

A l'appel, MADMACS définit quatre zones qui seront visualisées différemment en fonction de votre écran (fig 1):

- la zone de dialogue utilisateur-MADMACS pour un dialogue restreint à quelques lignes.
- la zone état pour des informations sur l'état du système. A l'appel cette zone contient la chaîne "MADMACS" suivi du numéro de version, puis le nom du tampon ouvert à l'appel dont le nom est par défaut "main".
- la zone graphique pour la visualisation du dessin de masques du circuit.
- la zone de texte pour des réponses nécessitant plusieurs lignes.

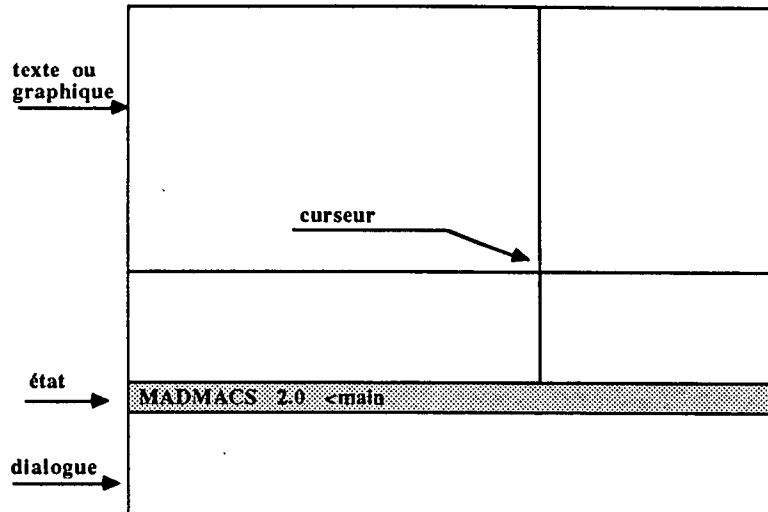


Figure 1: Configuration de l'écran sur Tektronix 4109

MADMACS n'utilise pas de menu avec souris, mais comme sous EMACS (éditeur de textes) le menu est attaché au clavier. Une touche ou une séquence limitée à quelques caractères (minuscules ou majuscules) correspondent à une commande du système. Les commandes sont mnémoniques (a : ajuster, + : zoom) et utilisent également les touches CTRL ou ESC (CTRL e : étendre). L'apprentissage est un peu plus compliqué que pour les systèmes avec souris et menus, mais l'expérience prouve que l'utilisation est ensuite plus rapide. La composition de commandes avec CTRL (noté ^) ou ESC doit s'interpréter comme suit :

^x ou CTRL x signifie tapez la touche CTRL et la touche x simultanément.

ESC x signifie tapez la touche ESC puis tapez la touche x.

La frappe d'une commande a un effet immédiat et ne nécessite en général pas de validation. Certaines des commandes demandant un complément d'informations dans la zone de dialogue (ESC u demande 'Facteur :'), se terminent par un retour chariot pour la validation. Le complément d'informations nécessaires pour l'exécution est de quatre types :

- caractère (ne demandant généralement aucune validation)
- entier positif (12 caractères au plus)
- chaîne courte (12 caractères au plus)
- chaîne longue (80 caractères au plus)

La nature du complément d'informations souhaité est précisé par un message dans la zone de dialogue. Les conventions pour les types de paramètres sont standards, un nom commence par une lettre,... Dans le cas de paramètre trop long le système tronque; attention dans la version actuelle aucun message ne signale une troncature éventuelle, les caractères en sus sont interprétés alors comme des commandes. Il est possible de corriger un paramètre en cours de saisie par les caractères # (suppression du caractère précédent) et @ (suppression de tous les caractères tapés).

Il est possible d'arrêter une commande lorsqu'elle demande un complément d'informations. Par exemple, tapez la commande "ESC u". Le système demande, dans la zone de dialogue, "Facteur :". Pour annuler cette requête en cours, tapez la commande "^g" (ce qui s'appelle "se taper la cloche" dicit P. Quinton). Notons cependant que l'utilisateur peut arrêter la commande uniquement si MADMACS demande des informations supplémentaires. Petit désagrément, l'utilisateur est quelque fois amené à taper deux fois la commande ^g pour arrêter la commande en cours.

MADMACS possède un aide-mémoire de commandes disponible en ligne à tous moments. Pour l'afficher dans la zone de texte, tapez la touche "?". Les commandes disponibles sont affichées à l'écran par fenêtres successives. Pour le défilement de ces fenêtres, tapez une touche de votre choix. Notons que l'on peut arrêter cette commande en cours par deux "^g" successifs.

Enfin pour sortir de MADMACS, utilisez la commande "q" (quitter). Dans la zone de dialogue, le système demande une validation à la commande tapée. Vous devez répondre :

Soit par o (pour oui) si vous désirez réellement sortir.

Soit par n (pour non) dans le cas contraire, ce qui vous permet d'effectuer vos sauvegardes.

Dans ce cas, répondez par n (ou tout autre caractère différent de o) pour rester sous le système MADMACS, et continuez les manipulations proposées.

1.3 Le curseur

Dans la zone graphique, vous voyez un curseur. Selon le type de terminal employé, il est matérialisé soit par un petit carré, soit par deux lignes blanches, une verticale à gauche de l'écran et l'autre horizontale en bas de la zone graphique (fig. 1: le curseur de MADMACS est matérialisé par l'intersection de ces deux lignes). Le curseur pointe toujours au centre d'un lambda carré (lambda : paramètre représentant la résolution du processus de fabrication [Mead 80]). De plus le curseur pointe sur un objet unique ou sur l'espace vide, comme on le voit par la suite.

2 Déplacements simples du curseur

Le premier jeu de commandes de déplacement du curseur est asservi à une grille dont le pas est fonction du paramètre λ et du facteur d'échelle courant. Le curseur se déplace d'un λ dans n'importe laquelle des 8 directions définies par les touches d'un pavé numérique.

7	8	9
4	5	6
1	2	3

Dessin du pavé numérique

- 1 : un λ vers le bas et un λ vers la gauche.
- 2 : un λ vers le bas.
- 3 : un λ vers le bas et un λ vers la droite.
- 4 : un λ vers la gauche.
- 6 : un λ vers la droite.
- 7 : un λ vers le haut et un λ vers la gauche.
- 8 : un λ vers le haut.
- 9 : un λ vers le haut et un λ vers la droite.

Le système MADMACS calcule automatiquement le facteur d'échelle pour la visualisation du circuit. A l'appel de MADMACS, le facteur est par défaut : un λ carré = un pixel (point de l'écran sur terminal graphique). Les déplacements du curseur sont donc peu perceptibles sur des écrans graphiques à haute résolution. Tapez plusieurs fois la même commande pour constater un déplacement du curseur ou bien augmentez le facteur d'échelle grâce à la commande de zoom : "+" (plus d'informations). L'opération inverse est effectuée par la commande de panoramique "-" (moins d'informations). L'utilisateur MADMACS peut grâce à ces 2 commandes contrôler la taille du dessin. Pour revenir au facteur d'échelle donné par défaut à l'appel, il suffit de taper la commande "ESC l".

Le déplacement du curseur λ par λ est fastidieux. Pour lui faire parcourir de grandes distances, MADMACS offre trois possibilités : soit appliquer un facteur multiplicatif à une commande, soit un positionnement absolu par coordonnées, soit un déplacement direct du curseur par un palet ou une souris.

Tapez la commande "ESC u". MADMACS demande : "Facteur :" (sous-entendu de répétition), dans la zone de dialogue. Répondez en lui donnant un entier positif 'n' validé soit par un retour-chariot (noté RC), soit par un espace (noté SP). Si l'entier n donné est de signe négatif, le système le force à 1. Si par contre vous donnez un n nul, la commande suivante ne sera pas exécutée (notez que répondre par "^g" force n à zéro). Cette notion est très utile pour les PIA présentés après (on peut ainsi contrôler l'exécution ou la non

exécution d'une commande). La commande suivante est exécutée n fois. Par exemple, ESC u 20 RC 6 déplace le curseur de 20 positions vers la droite. Vous ne pouvez pas appliquer le facteur multiplicatif à certaines commandes de MADMACS. Dans ce cas, MADMACS refuse la commande. Citons en exemple la commande vue précédemment "?", il n'y a pas de raison en effet de lister plusieurs fois de suite l'aide-mémoire.

L'utilisateur peut également demander un positionnement absolu par coordonnées. La commande "g" demande en paramètres le nombre de lambdas en x et le nombre de lambdas en y , puis positionne le curseur à l'endroit voulu. Quand vous appelez MADMACS ou ouvrez un nouveau tampon (cette notion est vue après), le curseur est positionné aux coordonnées (0,0). En fait, la commande g est rarement utilisée. Les autres jeux de déplacements du curseur la rendent inutile. Cette commande prend en paramètre deux entiers. Si vous ne passez pas de bonnes valeurs, le déplacement n'est pas pris en compte.

La troisième possibilité de déplacement du curseur utilise les notions de palet ou de souris offertes par certains systèmes. Ces interfaces permettent à l'utilisateur de déplacer le curseur à l'intérieur de l'écran. L'intégration de ces facilités dans Madmacs est traitée au coup par coup. Notons cependant l'utilisation du palet sur terminal Tektronix 4109. Le système Madmacs désactive ces facilités lors de la construction d'une macro ou d'un script. Le bien fondé de cette désactivation est explicité dans les paragraphes suivants.

Dans le pavé numérique, la touche "5" n'est pas utilisée pour des déplacements du curseur. Vous pouvez l'utiliser pour repositionner le curseur à sa position réelle. Certains systèmes permettent de déplacer le curseur par une souris ou un palet. Le positionnement du curseur perd de sa précision, il n'est pas replacé exactement au centre d'un lambda carré. La commande 5 permet dans ce cas un repositionnement sans ambiguïté.

3 Les objets manipulés

Les objets manipulés par MADMACS sont dessinés dans la zone graphique. Grâce au curseur, l'utilisateur peut décrire des distances et des dimensions. Dans ce paragraphe, nous nous attachons à la description des commandes de base pour le dessin d'un circuit. Sous MADMACS, vous manipulez deux types d'objets : des figures et des rectangles. Nous définissons plus en détail ces deux notions avant de passer aux commandes de création, de modification et d'effacement.

3.1 Cellule-Figure-Rectangle

Les objets de base manipulés par MADMACS sont toujours rectangulaires. On voit ici une simplification par rapport au langage CIF [Mead 1980] qui prévoit des structures de polygones et de cercles.

MADMACS intègre la notion de hiérarchie dans le processus de conception d'un circuit. Le concepteur peut créer les différents blocs composant le plan de masse de son circuit, il redéfinit successivement chaque bloc en plusieurs blocs pour arriver finalement au dessin complet de son circuit. Un circuit (ou un morceau de circuit) est décrit sous forme d'arbre.

Un rectangle est colorié et représente un morceau de couche technologique (une couleur est associée à chaque couche technologique). L'ensemble de tous les rectangles définit le dessin de masques du circuit. Dans la hiérarchie, un rectangle est considéré comme une feuille.

Une figure est un noeud de la hiérarchie. Elle est définie comme un ensemble de déclarations de rectangles et/ou de déclarations de figures. Les figures permettent de dessiner les blocs de différents niveaux du plan de masse du circuit. On peut définir par exemple un bloc PLA qui est la réunion des blocs PLAN-ET et PLAN-OU. Une figure est visualisée par un rectangle évidé de couleur marron. Lors de sa création, cette figure ne contient pas de sous-objets, nous verrons les manipulations très simples à effectuer pour donner un contenu à cette figure.

La notion de cellule est employée textuellement pour désigner indifféremment un rectangle ou une figure. Certaines commandes sont applicables indifféremment à un rectangle ou à une figure, on parle dans ces cas de commandes applicables à une cellule. On précise dans les autres cas le champ d'application des commandes.

3.2 Création d'une cellule

Toute cellule est rectangulaire et peut être définie par deux points diagonalement opposés. Pour la définition de ces points on prend pour l'un la position actuelle du curseur, et pour l'autre une marque (dite courante) mise préalablement.

3.2.1 La marque courante

La marque courante est un mécanisme permettant de mémoriser la position du curseur. Pour mettre cette marque, il suffit simplement de taper "." ; MADMACS répond dans la zone de dialogue "Set" pour signifier à l'utilisateur qu'il a bien mis la marque courante à l'emplacement du curseur. La mise de la marque courante n'apparaît pas dans la zone graphique. En cours de conception l'utilisateur mémorise aisément l'endroit où elle est.

Pour créer une cellule, mettez la marque courante sur un coin et déplacez le curseur vers le coin diagonalement opposé. La création d'une cellule de 10 lambdas par 5 lambdas nécessite après le positionnement de la marque courante, d'exécuter 9 fois la commande 6 suivi de 4 fois la commande 8. Vous avez ainsi défini les dimensions d'une cellule. Cette cellule peut être soit un rectangle, soit une figure. La figure 2 illustre la manipulation proposée ci-dessus.

3.2.2 Création d'un rectangle

Tapez la touche "c" (créer rectangle couche), vous voyez apparaître le rectangle colorié en marron. Le marron est la couleur indéfinie positionnée par défaut.

Pour créer des rectangles de couleurs différentes, changez de crayon de couleur en employant la commande "ESC c" (crayon). Le système demande la couleur choisie (message "Couleur:" dans la zone de dialogue). En technologie NMOS, répondez par d, p, m, i, c ou g (sans validation) qui correspondent aux différentes couches technologiques (diffusion en vert, polysilicium en rouge, métal en bleu, implant en jaune, contact en blanc et passivation en mauve). MADMACS met à jour la zone état en affichant après le numéro de version le caractère correspondant à la couche demandée. L'exécution de la commande 'ESC c p' affiche le caractère 'p' dans la zone état. La couleur courante est désormais la couleur affichée dans la zone état. L'utilisateur peut se tromper en passant une couche non existante dans la technologie employée. Dans ce cas MADMACS repositionne la couleur courante en marron (couleur par défaut). Dans la version actuelle, seule la technologie NMOS est valide. Très prochainement l'utilisateur pourra avoir accès à une quelconque technologie.

Vous pouvez créer plusieurs rectangles de couleurs différentes. Dans le cas de superposition de rectangles, un mélange est effectué pour la zone de recouvrement ce qui permet de voir les différentes couches de cette zone. L'utilisateur peut remarquer des exceptions à la règle donnée, des phénomènes d'absorption par une couleur sont possibles sur certains terminaux graphiques ne possédant pas assez de plans mémoire (exemple du contact (blanc) sur terminal Tektronix 4109).

3.2.3 Création d'une figure

Après le positionnement de la marque courante à un coin et du curseur au coin diagonalement opposé, l'utilisateur peut demander la création d'une figure. Tapez la touche "f" (créer figure). Dans la zone de dialogue, le système demande le nom (12 lettres au plus sinon remis à vide) à donner à la figure. Après réponse, vous voyez apparaître les contours en brun d'un rectangle correspondant à la figure créée.

Une figure est vide lors de sa création, cependant par définition elle est un ensemble de rectangles et/ou de figures. Pour définir les cellules contenues dans cette figure, il faut "descendre" dans celle-ci, on parle dans ce cas de descente dans la hiérarchie. Descendre dans une figure vide positionne l'utilisateur sur un écran vide symbolisant le contenu de la figure. Il doit alors définir le contenu de la figure. La notion de déplacements dans la hiérarchie est vue plus en détail au paragraphe 3.3.

3.2.4 La cellule pointée

Vous êtes désormais capable de dessiner tout un ensemble de cellules. Nous introduisons la notion de cellule pointée, de nombreuses commandes se basent sur cette notion comme nous le voyons par la suite. Déplacez le curseur sur une cellule. S'il n'y a pas d'ambiguïté de recouvrement, on dit que cette cellule est la cellule pointée. Les déplacements effectués à l'intérieur d'une cellule ne changent pas la cellule pointée, même lorsque le curseur est placé sur une zone de recouvrement.

Maintenant déplacez le curseur pour passer de l'espace vide à une zone de recouvrement: il y a ambiguïté sur la cellule pointée. Pour lever cette ambiguïté, faisons d'abord la remarque suivante: dans presque tous les cas, un recouvrement n'est pas total. Il est donc toujours possible de déplacer le curseur sur une partie de la cellule désirée non recouverte afin de la pointer sans ambiguïté. Si ensuite vous déplacez le curseur par les commandes 1..9 sur une zone de recouvrement sans sortir de la cellule pointée, la cellule pointée reste inchangée. Dans le contexte d'une zone de recouvrement où deux cellules uniquement sont concernées, vous pouvez pointer l'autre en exécutant simplement la commande "A" (autre cellule).

Pointer sans ambiguïté une cellule dans le cas d'un recouvrement total peut être résolu grâce à d'autres commandes. Notons que ce problème ne concerne que les rectangles. Un recouvrement est nécessairement entre rectangles de couleurs différentes. Un ensemble de commandes s'applique sur le rectangle de couleur courante et permet de résoudre toutes les actions à effectuer sur une zone de recouvrement. Exemple la commande "^x." positionne la marque courante sur le rectangle de couleur courante (s'il existe).

3.3 La manipulation de la hiérarchie

Un circuit est décrit sous forme d'un arbre où les figures sont les noeuds et les rectangles les feuilles. L'utilisateur peut donc demander à se déplacer dans cet arbre et effectuer des modifications à différents niveaux. Notons qu'à un instant donné, vous êtes positionné sur un père unique et seul ses fils sont visualisés.

3.3.1 Les déplacements dans la hiérarchie

Les figures créées sont toutes vides. Il convient de leur donner un contenu. Pointez la figure de votre choix. Tapez la commande ":". MADMACS efface votre écran, et vous donne ainsi un nouvel espace de travail. Lors de la création de la figure les seuls renseignements donnés ont été ses dimensions et son nom. Dans cet espace, l'utilisateur crée le contenu de la figure, mais en se détachant, pour plus de commodité, du contexte dans lequel elle est placée. Une figure peut être répliquée et placée dans des contextes différents.

Après définition de la figure, on peut remonter au niveau supérieur en tapant la commande ",,". On retrouve ainsi l'espace initial et le contexte d'emploi de la figure. Si vous êtes sur la

racine du circuit, il est bien entendu que la commande `,"` n'est plus valide. Un essai pour remonter entraîne la diffusion d'un message d'erreur dans la zone de dialogue : "Vous êtes sur la racine". Les deux commandes `,"` et `:"` permettent un déplacement rapide dans la hiérarchie.

Revenons sur la commande `,"`. Première constatation, la taille de la figure a peut-être changé. En effet à la suite de modifications dans la figure, MADMACS recalcule automatiquement la boîte englobant tous les objets contenus dans une figure. Donc la taille peut soit augmenter, soit diminuer. Deuxième constatation, la figure est toujours représentée par un rectangle évidé aux contours bruns. Vous pouvez cependant avoir des informations sur son contenu (s'il existe).

3.3.2 Les informations graphiques sur les figures

A tout moment, le concepteur peut demander les détails de la figure pointée grâce à la commande `"^z*`". MADMACS affiche tous les rectangles contenus dans la figure pointée et dans les figures de plus bas niveau. Les rectangles dessinés ne sont que des informations graphiques non manipulables. La complexité de la figure à détailler peut entraîner un temps de réponse assez long (voir très long!), il est à la charge du concepteur d'évaluer la complexité de cette figure et par conséquent le bien fondé de l'emploi de cette commande. Notons également la commande `*` qui détaille toutes les figures visibles à l'écran.

MADMACS peut également visualiser les connecteurs des figures par la commande `"$"`. Quand un rectangle de la figure (ou d'une sous-figure) a un bord commun avec la figure, il est découpé pour donner naissance à un connecteur qui a une épaisseur de 1 lambda. Dans la version actuelle, les connecteurs sont donc créés automatiquement par le système. Seules les opérations de visualisation et de déplacement à un connecteur sont possibles. Cependant il est envisagé d'avoir des objets connecteurs manipulables dans la version ultérieure de MADMACS. Le déplacement à un connecteur est détaillé dans un paragraphe suivant. Pour annuler la visualisation systématique des connecteurs, exécutez de nouveau la commande `"$"`.

3.3.3 La modification de la hiérarchie

Avec les commandes présentées ci-dessus, la hiérarchie que vous avez créée est fixe. Deux commandes essentielles vont vous permettre de la modifier à volonté.

La première commande permet de regrouper un ensemble de rectangles et de figures dans une même figure. Placez la marque courante et déplacez le curseur à un coin diagonalement opposé. Vous définissez ainsi les dimensions d'une nouvelle figure. Toutes cellules comprises dans cette zone peuvent être "absorbées" dans cette nouvelle figure. Tapez la commande `"^xc"`, MADMACS demande "Nom de figure :", auquel vous devez répondre par un nom de 12 caractères au plus. Outre la construction de cette nouvelle figure MADMACS efface toutes les

cellules absorbées. Si vous descendez dans la figure créée, vous retrouvez les cellules effacées.

Cet effacement n'est pas toujours visualisé correctement. Pour le rafraîchissement de l'écran, utilisez la commande "^ap" qui efface tout l'écran et réaffiche le dessin en gardant le même facteur d'échelle. Notons l'exécution de la commande "p" qui efface également tout l'écran, mais recalcule le facteur d'échelle pour visualiser toutes les cellules en utilisant tout l'écran.

Nous avons vu comment ajouter un noeud dans la hiérarchie et palier à de petits désagréments graphiques. L'opération inverse aplanit la hiérarchie, en d'autres termes enlève un noeud en remontant ses fils d'un niveau. Déplacez le curseur sur une figure (non vide de préférence) et exécutez la commande "^aa". La figure pointée est détruite et MADMACS dessine tous ses fils à l'écran. Toutes les cellules ramenées sont alors accessibles à ce niveau.

3.4 Destruction d'une cellule

Vous avez créé un ensemble de cellules. Des modifications doivent être apportées et notamment la suppression de certaines cellules. Les commandes de destruction de cellules agissent sur la cellule pointée. Dans le cas d'une figure pointée, la destruction de celle-ci se fait par la commande "^af" (notons que la création se fait par "f"). Dans le cas d'un rectangle pointé, la destruction de celui-ci se fait par "^ac" (notons que la création d'un rectangle se fait par "c").

Le cas d'un recouvrement total de rectangles pose des problèmes. Deux rectangles, de métal et diffusion, se recouvrent totalement. Il y a ambiguïté sur la cellule pointée : rectangle de métal ou de diffusion. Pour détruire, sans risque d'erreur, le rectangle de métal, par exemple, positionnez la couleur courante à métal (commande ESC c m) et tapez la commande "^ak". Le rectangle de métal est alors détruit.

Lors des destructions, l'utilisateur peut remarquer que dans certains cas il y a des problèmes de rafraîchissement de l'écran : cas déjà évoqué précédemment d'une couleur qui absorbe les autres, et également cas de recouvrements partiels de deux rectangles de même couleur où la destruction d'un des deux efface la partie de recouvrement. Nous rappelons les commandes "p" et "^ap" (présentées avant) pour peindre ou repeindre votre écran.

Vous pouvez effacer malencontreusement une cellule. Il est possible de la récupérer en exécutant la commande "^ay". Suite à une destruction, MADMACS mémorise la dernière cellule détruite. Le principe de cette commande est de ramener cette cellule en tenant compte d'un vecteur de translation défini par la marque courante et la position du curseur. L'utilisateur peut donc utiliser cette commande soit pour ramener la cellule détruite à la même place, soit pour la ramener à une place différente (dans une autre figure par exemple). L'usage de cette nouvelle commande nécessite quelques précautions, avant de ramener la dernière cellule détruite, MADMACS

regarde si la marque courante est positionnée. Dans ce cas, la cellule sera ramenée en subissant une translation de vecteur marque courante - curseur, dans le cas contraire le coin en bas à gauche de la cellule est placé sur le curseur.

Suite à une annulation de suppression, la cellule doit être remplacée sans subir de translation. Quelle que soit la position du curseur, tapez "." pour positionner la marque et tapez "^y". Le vecteur de translation sera nul et la cellule est donc remplacée à sa position initiale.

3.5 Modification du dessin

Avec les commandes précédentes, les modifications et constructions de dessins peuvent devenir laborieuses. Par exemple pour changer une cellule de place, vous devez actuellement détruire la cellule et la recréer à la place voulue, à moins d'utiliser la commande "^y" et d'appliquer le vecteur de translation marque courante - curseur avant de la ramener. Les commandes présentées dans les paragraphes suivants permettent d'effectuer aisément des modifications sur un dessin.

3.5.1 Modification de la taille d'un rectangle

Deux commandes permettent de modifier la taille d'un rectangle à volonté. Vous pouvez étendre la taille du rectangle pointé dans n'importe quelle direction, par exemple l'allonger vers la droite, ou vers le haut, ou bien encore vers le haut et la droite. Pour ce faire, mettez une marque sur le bord droit du rectangle pointé. Déplacez le curseur vers la droite (hors du rectangle pointé) et tapez la commande "^e" (étendre). Le rectangle marqué est étendu vers la droite jusqu'au curseur. Pour étendre le rectangle vers le haut, il suffit de marquer le côté supérieur du rectangle, de déplacer le curseur vers le haut et de taper la commande "^e". Vous pouvez effectuer ces 2 opérations en simultané. Marquez le coin supérieur droit du rectangle. Déplacez le curseur vers le haut et vers la droite. Tapez la commande "^e", le rectangle est étendu vers le haut et vers la droite jusqu'au curseur. La figure 3 illustre cette manipulation de rectangle et l'utilisation du vecteur marque courante - curseur.

Maintenant présentons le mécanisme exact de l'extension d'un rectangle. Vous positionnez une marque n'importe où sur le rectangle pointé. Déplacez le curseur où vous le désirez. Vous avez défini un vecteur marque courante-curseur. Supposons que ce vecteur soit (2,-3) dans le repère (gauche-droite,bas-haut). A l'exécution de la commande "^e", le système en déduit d'étendre le rectangle marqué de 2 lambdas vers la droite, et de 3 lambdas vers le bas.

La deuxième commande permet de diminuer la taille d'un rectangle. Le principe d'exécution est similaire à l'extension. Mettez une marque sur un rectangle, déplacez le curseur afin de définir un vecteur marque courante - curseur. Tapez la commande "s", MADMACS déduit la (ou les) direction vers laquelle le rectangle doit être diminuer et de combien. Si un rectangle est extensible à l'infini,

par contre il y a des limites pour le diminuer. Les nouvelles dimensions du rectangle seront au moins d'un lambda par un lambda dans le cas où l'utilisateur demandent des diminutions supérieures à la taille du rectangle.

3.5.2 Déplacement de cellule

Pour déplacer une cellule, le mécanisme est semblable à celui de la commande "^e". Il suffit de marquer une cellule de votre choix, puis de déplacer le curseur dans une direction quelconque et tapez la commande "a" (ajuster). En fonction du vecteur défini par la marque et le curseur, MADMACS translate la cellule dans la direction indiquée et selon la distance donnée. La figure 5 donne un exemple d'ajustement de deux figures, dans ce cas leurs contenus sont visualisés pour mettre en évidence les connections.

3.5.3 Orientation de cellule

Le curseur est positionné sur une cellule de votre choix. A cette cellule vous pouvez faire subir une rotation de + 90 degrés ou - 90 degrés, ou alors une opération de miroir par rapport à l'axe des x ou l'axe des y.

- "/" fait subir à la cellule pointée une rotation de + 90 degrés.
- "\" fait subir à la cellule pointée une rotation de - 90 degrés.
- "h" (horizontal) fait subir à la cellule pointée un miroir sur les x.
- "v" (vertical) fait subir à la cellule pointée un miroir sur les y.

Suite à une orientation d'une figure non vide, le détail de la figure fait apparaître son contenu orienté. La figure 4 illustre sur un exemple simple des orientations possibles d'une figure.

3.5.4 Reproduction d'une cellule

Revenons sur la commande "^y". Suite à une destruction, la cellule peut être ramenée. Ce mécanisme tient compte de la valeur du vecteur de translation marque courante - curseur pour traduire la cellule. L'utilisateur peut appliquer plusieurs fois cette même commande, donc obtenir plusieurs instanciations de la cellule. Entre deux exécutions, vous avez loisir de modifier la position du curseur et par conséquent la valeur du vecteur de translation. Les cellules ainsi reproduites sont toutes identiques, elles diffèrent par leur position. Un emploi judicieux de cette commande permet la construction aisée de structures régulières.

La commande "^y" ramène la dernière cellule détruite (méorisée par le système). Il est cependant possible de mémoriser une cellule particulière sans la détruire. La commande "ESC w" mémorise la cellule pointée. Une utilisation de la commande "^y" réplique alors la cellule pointée.

La réplique d'une cellule se fait souvent en plaçant la nouvelle instanciation juste à côté. C'est fréquemment le cas dans les structures régulières. Les commandes présentées maintenant permettent

de répliquer la cellule pointée dans les 8 directions données par le pavé numérique. Par exemple, la commande " $\wedge x6$ " crée une instanciation de la cellule pointée qui est ajustée à droite. Notons que cette nouvelle instanciation est la nouvelle cellule pointée. L'exécution de la commande " $\wedge x1$ " réplique la cellule pointée de façon à faire coïncider le point bas-gauche de la cellule répliquée et le point haut-droit de la nouvelle instanciation. Dans tous les cas d'utilisation, la nouvelle instanciation devient la cellule pointée.

Désormais vous êtes capable de répliquer à volonté une cellule et donc de construire aisément des structures régulières. Descendez dans une figure, effectuez des modifications puis remontez au niveau supérieur. Si vous demandez à voir tous les détails, vous constatez que toutes les figures sont modifiées. Une réplique de figure est une instanciation, non une création. Ce mécanisme permet de modifier des structures régulières rapidement en modifiant seulement les cellules de bases (attention cependant aux dimensions des figures qui peuvent changées, et donc aux connections inter-cellulaires).

3.5.5 Changer la couleur d'un rectangle

En cours d'édition d'un circuit, il peut arriver de se tromper de couleur lors de la création d'un rectangle. Nous présentons ici une commande permettant de modifier la couleur du rectangle pointé. Supposons qu'il soit en métal au lieu d'être en diffusion. Première action, changez la couleur courante pour la positionner à diffusion. Maintenant exécutez la commande " M " qui donne au rectangle pointé la couleur courante.

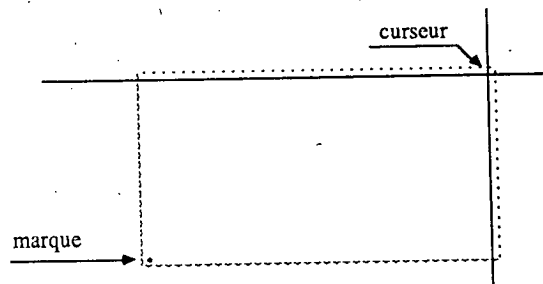


Figure 2: Création d'une cellule

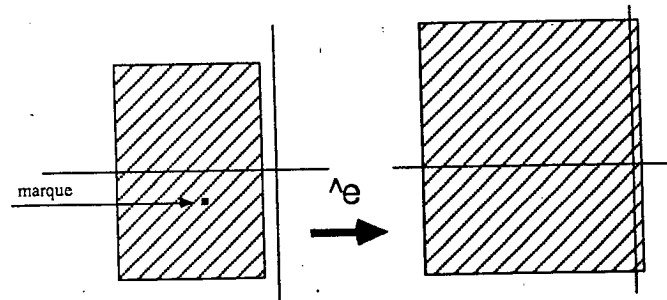


Figure 3: Etendre un rectangle.

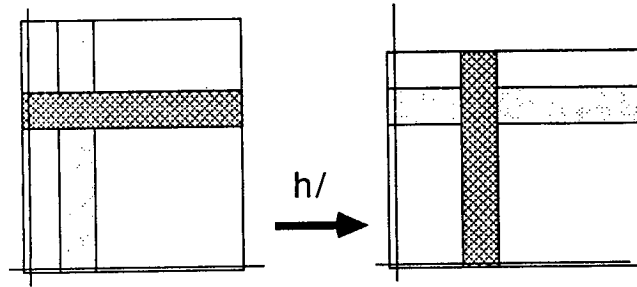


Figure 4: Orientation d'une cellule

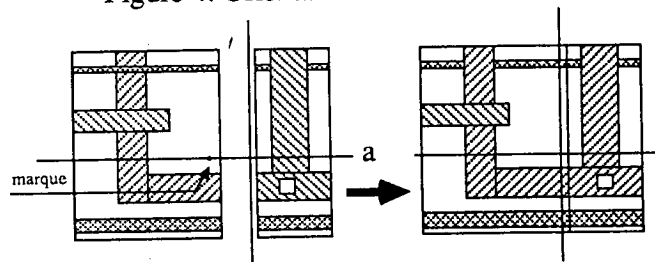


Figure 5 : Ajuster la position d'une cellule

Figures 2 à 5: Modifications de cellules

4 Les déplacements relatifs du curseur

Nous présentons dans ce paragraphe un nouveau mode de déplacement qui confère à MADMACS son originalité. Le système connaît à tout moment (ou est capable de retrouver) la topographie du dessin visualisé. L'idée est de se servir de ces connaissances pour se déplacer plus rapidement et également plus précisément. Nous avons développé un jeu de commandes qui permettent à l'utilisateur des déplacements où seule la direction est spécifiée. Il n'a plus à se préoccuper des distances.

Vous êtes actuellement positionné sur une cellule quelconque. Vous désirez vous déplacer sur la cellule immédiatement à sa droite et distante de 10 lambdas par exemple. Vous devez donc exécuter au moins 10 fois la commande 6 pour déplacer le curseur dans cette direction. MADMACS connaît les coordonnées de toutes les cellules du dessin. Il peut donc trouver la cellule immédiatement à sa droite et par conséquent calculer le déplacement que doit effectuer le curseur. On parle dans ce cas de déplacement relatif. Le concepteur ne se

préoccupe plus des distances et des dimensions dans son dessin. MADMACS est un système où les déplacements sont indépendants d'un système de coordonnées fixes. Notons que les coordonnées exactes sont gérées par le système.

Ces déplacements sont interactifs et surtout rapides. Plusieurs jeux de commandes sont mis à la disposition de l'utilisateur.

4.1 Déplacement inter-cellulaire

Ces déplacements permettent de déplacer le curseur de la cellule pointée à l'autre immédiatement voisine et positionnée sur le même axe. Rappelons que l'écran du terminal employé est considéré comme une fenêtre sur le dessin en cours. Par conséquent seule une partie du dessin peut être visible. Cependant les déplacements peuvent se faire à des cellules non visibles (présentes en dehors de l'écran), dans ce cas MADMACS recadre automatiquement l'écran sur la nouvelle cellule pointée, en plaçant le curseur toujours au centre de l'écran. Notons que l'exécution d'un déplacement sans succès génère le message d'erreur : "Pas de cellule dans cette direction". En cas de succès, la cellule atteinte devient la nouvelle cellule pointée.

- u (up) déplacement à la cellule immédiatement au dessus.
- d (down) déplacement à la cellule immédiatement au dessous.
- l (left) déplacement à la cellule immédiatement à gauche.
- r (right) déplacement à la cellule immédiatement à droite.

4.2 Déplacement intra-cellulaire

Ces déplacements permettent de positionner le curseur sur un des côtés de la cellule pointée. Cependant dans le cas où le curseur est positionné sur le côté demandé dans la cellule pointée, MADMACS déplace le curseur sur le côté désiré de la cellule immédiatement voisine (dans le cas où elle existe). Dans ce cas, la cellule atteinte devient la nouvelle cellule pointée, sinon elle est inchangée. Cet effet sera abandonné dans la prochaine version, suite aux sollicitations des utilisateurs.

- U (up) déplacement en haut de la cellule pointée.
- D (down) déplacement en bas de la cellule pointée.
- L (left) déplacement à gauche de la cellule pointée.
- R (right) déplacement à droite de la cellule pointée.

Vous êtes positionné sur un côté de la cellule et désirez positionner le curseur sur le côté opposé sans vous soucier de la direction exacte. Nous introduisons ici la notion de déplacement suivant un axe indépendamment de la direction. Le calcul de la direction est assuré par le système.

- i déplacement au côté opposé suivant l'axe des X.
- j déplacement au côté opposé suivant l'axe des Y.

4.3 Alignement par rapport à une autre cellule

Tous les déplacements déjà vus se passent entre cellules ou dans une cellule. Le jeu de commandes présenté dans ce paragraphe permet des alignements par rapport aux cellules du voisinage, mais ne pouvant être directement atteintes par les commandes de déplacement vues ci-dessus. Prenons l'exemple d'un alignement vers la droite (le mécanisme est similaire pour les autres alignements). L'exécution de la commande "`^r`" cherche la cellule dont le point bas-gauche est le plus près vers la droite. En cas de succès, l'abscisse du curseur devient l'abscisse du point bas gauche de la cellule répondant aux spécifications.

- `^u` (up) alignement à la cellule du dessus la plus proche.
- `^d` (down) alignement à la cellule du dessous la plus proche.
- `^l` (left) alignement à la cellule de gauche la plus proche.
- `^r` (right) alignement à la cellule de droite la plus proche.

4.4 Déplacement à un connecteur

Dans un paragraphe précédent, nous évoquons l'existence des connecteurs d'une figure. Il y est dit notamment qu'ils ne sont pas considérés comme des objets. Cependant des commandes de déplacements à ces connecteurs existent et facilitent la connection de fils à une figure préexistante. Visualisez les connecteurs de la figure pointée (commande "`$$`"), puis positionnez le curseur sur un des côtés de cette figure. Si vous êtes sur le côté gauche ou droit, vous pouvez accéder au connecteur supérieur en exécutant la commande "`ESC 8`", ou encore au connecteur inférieur par "`ESC 2`". Dans le cas d'un positionnement sur le côté supérieur ou inférieur l'accès au connecteur de gauche se fait par "`ESC 4`" et celui de droite par "`ESC 6`".

Les différents déplacements présentés ci-dessus permettent à l'utilisateur de se détacher complètement du système de coordonnées traditionnellement utilisé dans les éditeurs graphiques. La rapidité d'exécution est le premier point fort, le deuxième est une précision absolue du déplacement que l'utilisateur ne peut obtenir qu'avec un système de positionnement par coordonnées.

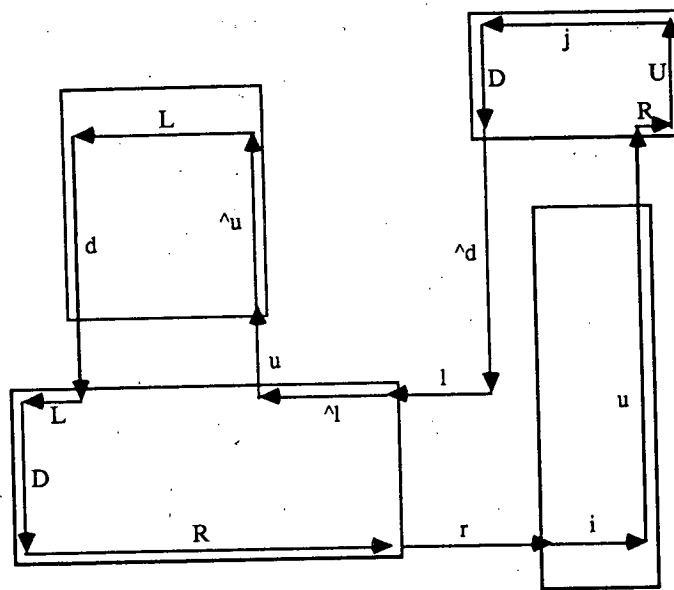


Figure 6: Utilisation des déplacements relatifs

5 Les marques

Une marque permet de mémoriser les coordonnées du curseur. MADMACS offre la possibilité d'avoir plusieurs marques nommées. Ces marques permettent uniquement la mémorisation de coordonnées. Leur mécanisme est différent de celui de la marque courante qui permet de mémoriser en sus la cellule pointée.

5.1 La marque courante

Pour la création d'une cellule nous avons vu précédemment l'utilisation de la marque courante. Rappelons que le positionnement de la marque courante se fait par la commande ".". MADMACS en plus du mécanisme de mémorisation de la position du curseur et de la cellule pointée offre des commandes permettant des alignements par rapport à cette marque ou des déplacements à cette marque. L'exécution de la commande " $\wedge x \wedge x$ " inverse la position du curseur et de la marque courante. Cette commande permet de se déplacer sur son circuit (tout en perdant la vision de l'endroit marqué) et de revenir exactement à la position initiale (si la marque courante est positionnée).

L'utilisateur peut tout à loisir s'aligner par rapport à la marque courante selon le même principe des alignements par rapport aux cellules. La commande " $\wedge x h$ " aligne le curseur sur la marque courante suivant l'axe des x. En d'autres termes le curseur prend la même abscisse que la marque, mais son ordonnée reste inchangée. La commande " $\wedge x v$ " a le même effet, mais suivant l'axe des y.

La marque courante n'est pas toujours positionnée. Après une création de cellule, par exemple, elle est effacée. Dans ce cas, si vous essayez d'exécuter une commande tenant compte de la marque courante, MADMACS n'exécute pas la commande et signale dans la zone de dialogue : "Pas de marque de mise".

5.2 Les marques nommées

Avoir une seule marque possible n'est pas concevable dans un système. Outre la marque courante, le concepteur peut positionner des marques nommées. Ces nouvelles marques permettent uniquement de mémoriser les coordonnées du curseur à un moment donné. Pour placer une marque nommée, exécutez la commande " $\wedge z$ ". MADMACS demande le nom de la marque employée (chaîne courte de 12 caractères au plus). Après déplacement du curseur, l'utilisateur peut à loisir le ramener à la place marquée en tapant la commande " $\wedge zg$ " suivi du nom de la marque.

Les marques nommées offrent également des possibilités d'alignement comme dans le cas de la marque courante. La commande " $\wedge x^h$ " demande le nom de la marque nommée avant de s'aligner suivant l'axe des x (si, bien sûr, cette marque existe). Son symétrique pour l'axe des y est la commande " $\wedge x^v$ ".

La figure 7 donne une illustration des différents déplacements et alignements possibles par rapport aux différents types de marques. Cette notion de marquage est très importante et très fréquemment utilisée pour automatiser la construction de morceaux de circuits. L'utilisation du marquage est souligné dans les paragraphes suivants.

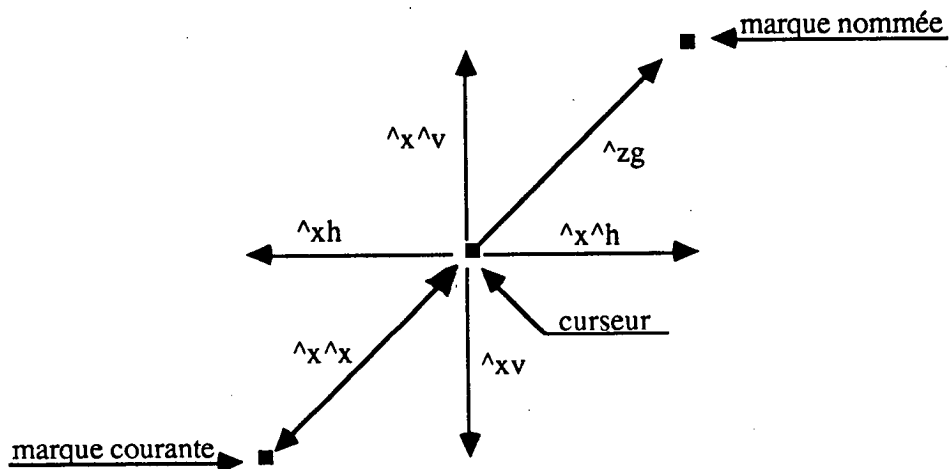


Figure 7 : Déplacements par rapport aux marques

6 Le nommage des objets

Lors de la création d'une figure, MADMACS demande son nom (chaîne courte). Cependant, tous les objets manipulables (figures et rectangles) peuvent être nommés et les noms modifiés à souhait par les utilisateurs.

6.1 Nommage d'une figure

Toutes les figures ont déjà un nom donné lors de leur création. L'utilisateur peut demander à lister dans la zone texte les noms de toutes les figures du circuit en cours de conception. La commande "p" liste, par fenêtres de 20 lignes, tous les noms de figures précédés de leurs numéros. La gestion des numéros de figures est faite par le système, et l'utilisateur n'y a pas accès. Notons que la figure racine du circuit a pour numéro par défaut 0. Le nom d'une figure peut être éventuellement vide, dans ce cas seul le numéro est listé.

La liste des noms de figures peut être très longue (car dépendante de la complexité du circuit). Si vous ne désirez que le nom de la figure pointée tapez la commande "^zp" qui vous donne la réponse voulue dans la zone de dialogue. Suite à cette réponse, vous pouvez changer le nom de la figure pointée en exécutant la commande "^zc" et lui donner un nouveau nom (éventuellement vide).

6.2 Nommage d'un rectangle

MADMACS offre également la possibilité de nommer les rectangles. Ce nommage présente de nombreux avantages, car il permet une description plus complète d'une cellule de base. De plus les noms de rectangles permettent de nommer un noeud électrique et donc d'avoir un résultat d'extraction de circuit électrique plus clair. (L'extracteur de circuits électriques est présenté plus loin). L'utilisateur peut associer au rectangle pointé un nom par la commande "n". Le nom est demandé dans la zone de dialogue et est une chaîne courte. Cependant le nom du rectangle pointé peut être changé grâce à la commande "N", ou bien encore détruit par la commande "^n". Changer ou détruire un nom de rectangle inexistant n'a aucun effet. En cours de construction du dessin, l'utilisateur peut oublier le nom associé à un rectangle. Pour demander à visualiser le nom du rectangle pointé dans la zone de dialogue, exécutez la commande "ESC n". Dans le cas où il n'y en a pas, MADMACS répond par une chaîne vide.

MADMACS ne garantit pas l'unicité des noms, il laisse à l'utilisateur le bon soin de la gestion des noms. De plus aucune commande (à part celles présentées ci-dessus) ne tient compte des noms de cellules. Actuellement, ces noms sont uniquement des informations de nature à commenter un circuit.

7 Les commandes de visualisation

Dans les paragraphes précédents, nous avons présenté quelques unes des commandes de visualisation disponibles sous MADMACS (+, -, ESC 1,...). Ces commandes permettent de contrôler la visualisation du dessin en cours d'édition. Nous présentons dans ce paragraphe d'autres possibilités.

7.1 Recadrage du dessin

Le curseur se trouve suite à des déplacements près d'un bord de l'écran. L'utilisateur peut recadrer tout le dessin en le ramenant de façon à placer le curseur au centre de l'écran. Il suffit de taper la commande "=" . Cette commande permet de contrôler le cadrage du dessin calculé par défaut. Vous pouvez ainsi déplacer aisément votre écran comme une fenêtre sur le dessin en cours, mais en gardant le curseur au centre de l'écran. Ceci permet de voir l'environnement autour du curseur, et donc favorise une meilleure utilisation des déplacements relatifs.

7.2 Recadrage du curseur

Les commandes présentées dans ce paragraphe peuvent être assimilées à des déplacements. Lors des déplacements présentés jusqu'alors la destination du curseur est connue et précise. Dans ce cas, la destination atteinte n'est pas connue précisément avant l'exécution. Ces commandes reposent toutes sur les coordonnées de la figure englobant toutes les figures et rectangles visualisés. Nous l'appelons la figure père. Elles permettent de déplacer le curseur, ou encore de le cadrer, dans une région donnée.

La commande ";" permet de recadrer le curseur aux coordonnées du point en bas à gauche de la figure père. Pour comprendre l'utilité de cette commande, imaginez le cas où plus rien n'est affiché à l'écran. Deux choses sont susceptibles d'être arrivées : vous êtes à un facteur d'échelle trop petit (suite à de nombreuses exécutions consécutives de la commande -), ou votre écran est positionné en dehors des limites de la figure père (ce qui arrive fréquemment lors des premières utilisations de MADMACS). Dans le premier cas, tapez la commande "P". Dans le deuxième cas, tapez la commande ";" qui recadre le curseur et donc permet de retrouver son dessin. Il est bon de souligner ici que la taille d'une cellule est à priori infinie, c'est à dire dépendante uniquement des capacités du système hôte (plus grand entier à plus petit).

Les commandes suivantes permettent de cadrer le curseur par rapport à la taille de la figure père. Elles sont utilisées généralement pour visualiser les bords de la figure père.

- ^zu (up) déplacement en haut de la figure père.
- ^zd (down) déplacement en bas de la figure père.
- ^zl (left) déplacement à gauche de la figure père.
- ^zr (right) déplacement à droite de la figure père.

7.3 Déplacement de l'écran par demi-fenêtres

Sur l'écran n'apparaît généralement qu'une partie du dessin du circuit. L'utilisateur souhaite voir les informations qui sont à droite. Les commandes suivantes permettent de visualiser le dessin en déplaçant l'écran demi-fenêtre par demi-fenêtre sur le dessin en cours d'édition :

[^]z[^]u (up) déplacement d'une demi-fenêtre vers le haut.
[^]z[^]d (down) déplacement d'une demi-fenêtre vers le bas.
[^]z[^]l (left) déplacement d'une demi-fenêtre vers la gauche.
[^]z[^]r (right) déplacement d'une demi-fenêtre vers la droite.

Suivant la remarque sur la taille des cellules faite précédemment, le déplacement par demi-fenêtre peut positionner l'écran sur une zone vide. L'utilisation de la commande ";" est alors utile.

7.4 Plan de masse

Dans les paragraphes précédents, nous avons vu les commandes "*" et "z*" qui permettent de détailler les figures en affichant les rectangles définis dans celles-ci. La commande "m" affiche le contenu des figures de façon similaire aux commandes * et ^z*, mais en limitant l'affichage aux sous-figures des figures visibles à l'écran. Les informations affichées ne sont pas manipulables à ce niveau. Pour avoir les seules cellules manipulables, rafraîchissez l'écran. Le temps d'exécution de la commande m est fonction de la complexité des figures affichées, les délais peuvent donc être longs.

Une figure ou un rectangle peuvent être orientés de différentes manières. Le détail ou le plan de masse d'une figure tiennent compte de ses éventuelles orientations. En exemple, répliquez une figure non vide. Orientez la deuxième instanciation (miroir suivant l'axe des x par exemple) puis demandez le détail de ces deux figures, vous constatez alors que le contenu des figures est identique mais orienté différemment. Si la figure est complexe (composée de sous-figure), vous remarquerez également les orientations en demandant le plan de masse.

7.5 Restriction de l'affichage

Des phénomènes d'absorption peuvent apparaître au cours de l'affichage, occultant ainsi les rectangles recouverts. La commande "v" permet de réafficher le dessin mais en ne visualisant uniquement que les rectangles correspondant à la couleur courante et les figures. Il est possible de travailler à ce niveau de visualisation (seuls les rectangles de diffusion sont dessinés par exemple). Mais attention, si les autres rectangles sont invisibles ils n'en sont pas moins présents. Cette commande doit donc être suivie, par précaution, par un "p" après consultation pour éviter des erreurs.

8 Les tampons

Au cours d'une session de travail, le concepteur peut désirer travailler sur un morceau spécifique de son dessin tout en conservant en ligne le dessin complet du circuit ou encore charger une librairie de cellules prédéfinies. En technologie NMOS, il est intéressant d'avoir une librairie contenant les différents contacts ou les pull-up idiomatiques.

8.1 Ouverture d'un buffer

MADMACS propose plusieurs tampons accessibles. Dans chaque tampon (ou buffer), vous pouvez charger un circuit différent ou des morceaux d'un même circuit.

Par exemple, vous désirez modifier une figure sans vous préoccuper du reste du circuit. Prenez cette figure pointée (par ESC w) et changez de buffer par la commande "**^xb**" suivi d'un nom (chaîne courte). Vous placez la figure dans ce nouveau buffer avec "**^y**", ce qui permet de travailler plus soupagement. Vous gagnez en rapidité d'affichage (uniquement la figure) et seules les informations utiles apparaissent. L'utilisation de "**^y**" permet d'avoir un pont entre les différents buffers. Lorsque vous ramenez une figure par "**^y**" dans un autre buffer, MADMACS vérifie si une figure possède le même nom. Dans ce cas, il demande à l'utilisateur s'il doit répliquer cette figure déjà existante ou en créer une nouvelle. Ce mécanisme permet de minimiser la taille du code d'un circuit.

Une autre utilisation possible consiste à charger en ligne une bibliothèque de cellules prédéfinies dans un buffer spécifique et de venir chercher dans celui-ci les cellules utiles pour le dessin du circuit.

NOTE : par défaut à l'appel de MADMACS, le tampon de travail porte le nom **main**.

Vous avez créé plusieurs buffers. Il est possible d'afficher les noms des buffers utilisés dans la zone de texte par la commande "**^xb**". Les noms de buffers précédés par une * spécifient les buffers modifiés (cette étoile en début de la zone état spécifie que le buffer courant est modifié). Notons que dans cette même zone, MADMACS affiche le nom du buffer courant et celui du précédent. Si vous exécutez la commande "**^xb**" et tapez directement un retour-chariot, MADMACS repositionne l'utilisateur dans le buffer précédent.

8.2 Gestion des buffers

Lorsque vous avez fini de travailler avec un buffer, vous pouvez le détruire en exécutant la commande "**^xk**" qui prend en paramètre un nom de buffer. Si aucun nom de buffer n'est donné, le buffer spécifié est alors le buffer courant et comme le système ne peut rester sur un buffer détruit, il demande alors le nom du nouveau buffer où il doit se positionner. Sans détruire le buffer, l'utilisateur peut demander

à vider le buffer de son contenu en tapant la commande "`^x^k`". L'utilisateur peut ainsi repartir sur un espace vide. Attention à l'emploi de ces deux commandes, aucune validation n'est demandée.

Vous êtes dans un buffer dont par mégarde vous venez de modifier le contenu. Vous restaurez le contenu dans son état initial, mais cependant le système notifie toujours que le contenu est modifié. Pour forcer l'indicateur buffer modifié à faux, il suffit d'exécuter la commande "`ESC ~`". Vous remarquez notamment la disparition de l'étoile dans la zone état.

9 Les scripts

Le mécanisme des scripts permet de mémoriser sur un fichier une série de commandes tapées au clavier, en parallèle à leur exécution. Les fichiers scripts sont ensuite réexécutables à volonté, mais dans le même contexte. L'utilisation la plus courante permet d'avoir un journal du travail effectué au cours d'une session de travail. Ou encore, il permet de mémoriser une série de commandes à exécuter plusieurs fois (cet avantage est peu utilisé car il existe également un mécanisme de macros présenté dans un paragraphe suivant).

Le mécanisme de script proposé par MADMACS permet d'ouvrir un script en mode apprentissage par la commande "`ESC o`" qui demande dans la zone de dialogue un nom de fichier (chaîne longue). Le nom de fichier passé en paramètre est celui du fichier qui contiendra la suite de commandes réexécutables. Pour toutes les commandes utilisant des fichiers, MADMACS effectue des contrôles pour vérifier l'existence du fichier et les accès autorisés. Dans le cas présent, le script est écrit sur un fichier. S'il existe déjà un fichier de même nom sans accès en écriture, le système annule la requête et sort du mode apprentissage script.

A tout moment, vous pouvez sortir du mode apprentissage script, et donc fermer le fichier script, par la commande "`^o`". L'utilisateur ne peut pas empiler les scripts, c'est à dire demander à passer en mode apprentissage script quand le système y est déjà. Dans ce cas, le système sort du mode apprentissage, ferme le fichier et notifie à l'utilisateur cette action.

Vous pouvez créer un petit fichier script, qui crée un rectangle de métal par exemple. La réexécution de cette série de commandes mémorisées se fait par la commande "`ESC i`" qui demande dans la zone de dialogue le nom du fichier script. Le système vérifie avant d'entrer en mode exécution script si le fichier demandé existe et les accès autorisés sont en lecture.

10 L'environnement système

Au cours d'une session de travail, un concepteur peut sauvegarder son dessin sur un fichier, charger un dessin déjà sauvegardé ou insérer dans le dessin en cours un morceau de dessin. Trois commandes adaptées sont disponibles pour effectuer ces opérations. Elles demandent par le biais de la zone dialogue le nom du fichier (chaîne longue). Le système vérifie les accès et l'existence du fichier demandé.

"^xw" pour sauver le dessin dans un fichier. L'étoile dans la zone état est effacée pour notifier que le contenu est sauvegardé.

"^xr" pour lire un fichier contenant un dessin. Le contenu du buffer est écrasé par le nouveau dessin chargé.

"^xi" pour insérer un fichier (dessin) à l'emplacement du curseur.

Ces trois commandes demandent en paramètre le nom du fichier contenant les informations désirées.

Un dialogue avec le système hôte est possible par le biais des commandes "^xm" et "^xe". Dans les deux cas, MADMACS demande une requête exécutable par le système hôte. La réponse sera dans la zone dialogue pour la commande "^xm" et dans la zone texte pour la commande "^xe". Pour dialoguer avec le système, l'utilisateur a donc le choix entre deux commandes. L'emploi de l'une ou l'autre commande se fait en fonction de la longueur de la réponse du système hôte. Une réponse ne comportant qu'une ligne peut être demandée par "^xm", tandis qu'une réponse de plusieurs lignes par "^xe".

Sous MADMACS, il est possible de faire de l'édition de textes. La commande "ESC e" appelle l'éditeur de textes du système hôte. Notons que sous MULTICS et SUN vous appelez Emacs, sous HP vi et sous SM90 Emin. Pour retourner sous Madmacs, il suffit à l'utilisateur de quitter l'éditeur. Le système Madmacs est automatiquement rappeler.

11 Outils de vérification NMOS

Deux outils de vérification pour la technologie NMOS (utilisant les règles définies par Mead et Conway) sont implémentés dans le système MADMACS. Le concepteur peut à tout moment appeler le vérificateur de règles de dessins ou l'extracteur de circuits électriques. Ces deux outils interactifs sont appelés de la même manière. Ils s'exécutent sur des fenêtres limitées à 100 x 100 lambdas. La zone est définie par la marque courante et la position courante du curseur.

Pour appeler l'extracteur de circuits électriques, il suffit de taper la commande "ESC x xtr", le résultat est soit visualisé à l'écran, soit mémorisé sur fichier. Dans les deux cas, le résultat

comprend une liste des noeuds électriques, puis une liste de transistors suivi d'un éventuel rapport sur les erreurs détectées. Les noeuds électriques sont numérotés automatiquement par le système. Cependant dans le cas d'un nommage d'un rectangle composant ce noeud électrique, le système associe au noeud le nom du rectangle. Dans le cas d'une double définition (deux rectangles du même noeud avec des noms différents) le rapport d'erreur signale que le premier nom est égal au deuxième nom. Nous soulignons ici l'importance du nommage des rectangles qui permet une meilleure lecture des résultats de l'extracteur et une possible détection d'erreur.

L'appel du vérificateur de règles de dessins se fait par "ESC x vrd". Le résultat donne le nombre d'erreurs, et pour chaque erreur détectée affiche la fenêtre erronée. On peut ensuite demander une justification des erreurs par la commande "ESC x jtf". Le vérificateur redessine une à une les fenêtres erronées en indiquant le type d'erreur détectée. Pour faire défiler les fenêtres, tapez un espace. Si vous tapez un autre caractère, vous sortez du mode de justification mais vous pouvez y rentrer à nouveau. Le rapport d'erreur n'est détruit que suite à une déconnexion ou à une autre demande de vérification. Les erreurs sont en général fonction du coin en bas à gauche de la fenêtre dessinée. Notons également qu'il est souvent préférable de rafraîchir l'écran entre le vérificateur et le justificateur pour effacer les fenêtres erronées. En annexe vous trouverez les types d'erreurs donnés par le vérificateur.

Ces deux outils sont essentiels en cours de conception des cellules de base ou en cours d'assemblage de cellules.

12 Les macros

Lors de la conception d'un circuit, certaines suites de commandes sont à appliquer plusieurs fois. MADMACS possède un mécanisme de macrocommande (appelée macro). Une macro est une suite de commandes mémorisées que l'on peut réexécuter à volonté sur différents morceaux du circuit. Ces macros sont construites interactivement. En entrant dans le mode macro-apprentissage par "^x(", toute commande tapée est exécutée normalement mais également mémorisée dans la macro courante, jusqu'à la sortie de ce mode par "^x)". En construisant la macro, le concepteur peut voir interactivement l'effet de la suite de commandes, il l'utilise uniquement si la macro est correcte de visu. Cette macro courante est réexécutable à volonté en tapant la commande "^xe".

La macro courante peut être sauvegardée dans le système en lui donnant un nom par la commande "^xn", ce qui permet d'avoir plusieurs macros valides dans le système. La réexécution d'une macro nommée se fait par "ESC x" suivi du nom de la macro donné dans la zone de dialogue. L'utilisateur vient ainsi de construire une nouvelle commande du système MADMACS.

Une macro peut appeler une macro. Seule restriction : nous interdisons strictement la récursivité. En cas de détection par le système, il essaie d'arrêter proprement, s'il ne le détecte pas alors

"Adieu Va". Par exemple, le système interdit de passer en mode apprentissage macro si on y est déjà.

Exemple de macro pour construire un carré :

```

^x(      Ouvrir la macro courante.
.        Mettre la marque courante.
ESC u 20 RC 9 Déplacer le curseur de 20 lambdas vers le haut et la
           droite.
c        Créer un rectangle.
^x)      Fermer la macro courante.

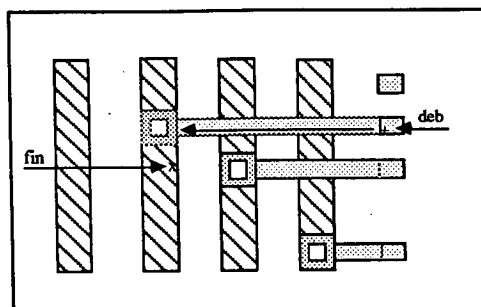
^xe      Réexécuter la macro courante.
^xn carre RC Donner le nom carre à la macro courante.
ESC x carre Exécuter la macro de nom carre.

```

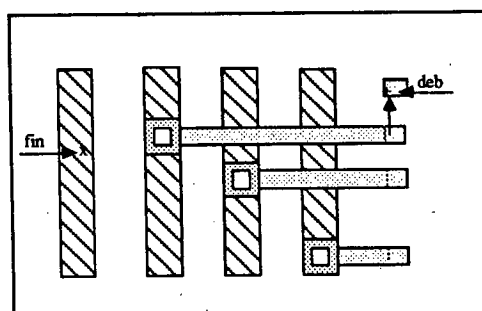
L'utilisation des macros permet de construire rapidement et avec plus de sécurité (due à l'interactivité de la construction) des séries de commandes réexécutable à volonté. La notion de macro associée aux déplacements relatifs permet de construire des macros très élaborées qui ont l'énorme avantage d'être en partie indépendante du contexte d'emploi. Par exemple nous définissons une macro qui se positionne sur le bord gauche d'une cellule, et crée un rectangle de ce bord à celui de droite. La réexécution de cette macro est indépendante de la taille de la cellule pointée, grâce notamment au déplacement relatif R (droite intérieur). Cet exemple met en évidence l'indépendance contextuelle citée ci-dessus. Les figures 8.10 montrent 3 exemples de macros pour le routage, qui peuvent être employées dans différentes parties du circuit. Ces macros étant construites par le concepteur, on peut parler de boîte à outils privée.

Dans la version actuelle, l'utilisateur ne peut ni éditer ses macros et ni les modifier (à moins de les reconstruire). Il peut cependant les visualiser dans la zone de texte. Pour la macro courante utilisez la commande "^x*", pour une macro nommée utilisez la commande "ESC x show-macro" suivi du nom de la macro à éditer.

Un utilisateur en cours de session peut créer tout un ensemble de macros. En fin de session, il a la possibilité de les sauvegarder en vue d'une utilisation ultérieure. La sauvegarde se fait sur fichier par la commande "ESC x save-macros" suivie d'un nom de fichier. Au début de la session suivante, le chargement se fait par la commande "ESC x load-macros" suivie du nom de fichier. Attention : si vous faites deux chargements successifs de fichiers de macros, seules les macros du deuxième fichier sont accessibles.



a)

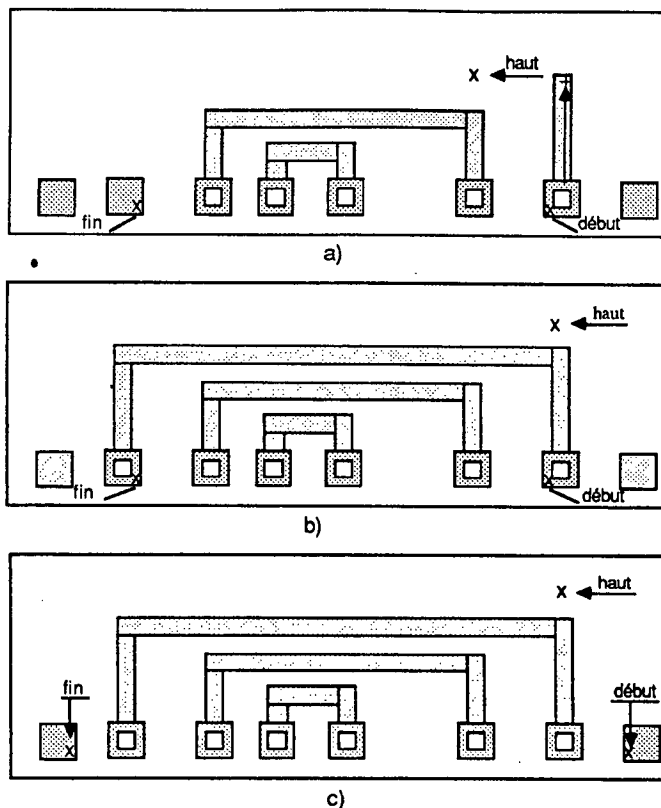


b)

Cette macro connecte des lignes horizontales alignées à un bus vertical. Avant exécution, l'utilisateur marque la première ligne du bus (fin) et la première ligne horizontale (deb). Ici, nous présentons les commandes de la macro (en littéral) et leur exécution. Deux exécutions ont eu lieu.

- a) Aller à la marque deb
Mettre la marque courante
Aligner horizontalement sur la marque fin
Déplacer le curseur d'1 lambda vers la droite
Etendre le rectangle marqué
Déplacer le curseur sur le côté gauche de la cellule de gauche
Déplacer le curseur d'1 lambda vers le bas
Insérer le fichier contact-diffusion-métal
- b) Déplacer le curseur à la cellule de gauche
Mettre la marque fin
Aller à la marque deb
Déplacer le curseur à la cellule au dessus
Mettre la marque deb

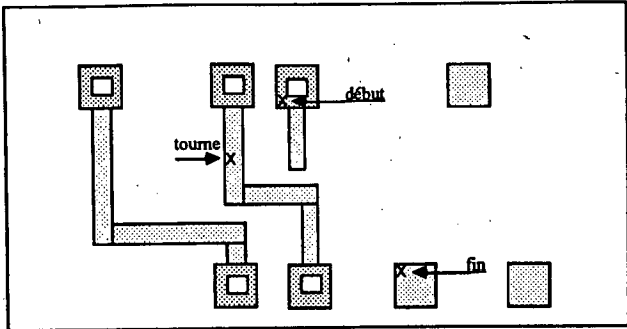
Figure 8 : Exemple de Macro de Connexion à un Bus



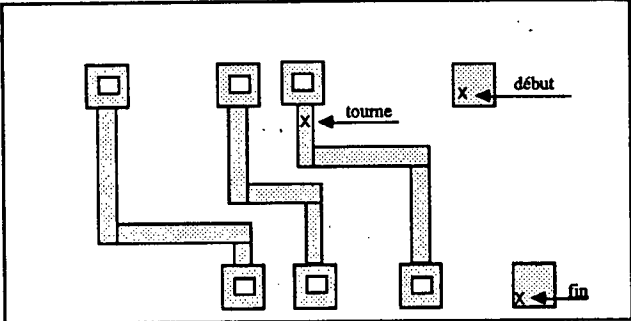
Cette macro connecte des contacts alignés. Avant exécution, l'utilisateur définit 3 marques: début, fin et haut qui indiquent le chemin de début à fin en passant par haut. Nous présentons les commandes de la macro (en littéral) et leur exécution. Deux exécutions ont eu lieu.

- a) Aller à la marque début
Insérer le fichier contact-diffusion-métal
Déplacer le curseur en haut de la cellule
Déplacer le curseur le curseur d'1 lambda vers le haut et la droite
Mettre la marque courante
Déplacer le curseur d'1 lambda vers la droite
Aligner verticalement sur la marque haut
Prendre le crayon diffusion
Créer un rectangle
- b) Déplacer le curseur de 6 lambdas vers le haut
Mettre la marque haut
Déplacer le curseur à la cellule en dessous
Mettre la marque courante
Déplacer le curseur d'1 lambda vers le bas
Aligner horizontalement sur la marque fin
Créer un rectangle
Déplacer le curseur d'1 lambda vers le haut et la gauche
Mettre la marque courante
Déplacer le curseur d'1 lambda vers la gauche
Déplacer le curseur à la cellule en dessous
Créer un rectangle
Aller à la marque fin
Déplacer le curseur de 3 lambdas vers la gauche
Insérer le fichier contact-diffusion-métal
- c) Déplacer le curseur à la cellule de gauche
Mettre la marque fin
Aller à la marque début
Déplacer le curseur à la cellule du dessus
Mettre la marque début

Figure 9 : Exemple de Macro Routage "Pont"



a)



b)

Cette macro connecte les contacts supérieurs (alignés) avec les contacts inférieurs (alignés). Avant exécution, l'utilisateur définit 3 marques : début, fin, tourne. Elles indiquent le chemin de début à fin en passant par tourne. Nous présentons les commandes de la macro (en littéral) et leur exécution. Deux exécutions ont eu lieu.

- a) Aller à la marque début
Insérer le fichier contact-diffusion-métal
Déplacer le curseur d'1 lambda vers la droite et le bas
Mettre la marque courante
Déplacer le curseur d'1 lambda vers la droite
Aligner verticalement sur la marque tourne
Prendre le crayon diffusion
Créer un rectangle
Déplacer le curseur de 5 lambdas vers le haut
- b) Mettre la marque tourne
Déplacer le curseur en bas de la cellule
Mettre la marque courante
Déplacer le curseur d'1 lambda vers le haut
Aligner horizontalement sur la marque fin
Déplacer le curseur d'1 lambda vers la droite
Créer un rectangle
Mettre la marque courante
Déplacer le curseur d'1 lambda vers la droite
Déplacer le curseur à la cellule en dessous
Déplacer le curseur d'1 lambda vers le haut
Créer un rectangle
Aller à la marque fin
Insérer le fichier contact-diffusion-métal
Déplacer le curseur à la cellule de droite
Mettre la marque fin
Aller à la marque début
Déplacer le curseur à la cellule de droite
Mettre la marque début

Figure 10 : Exemple de Macro de Routage Rivière

13 Les Procédures d'Implantation Automatique

Le mécanisme de macro pêche par l'absence de paramètres. Nous avons donc introduit un nouveau type de macros plus puissant car paramétrable. Quelques commandes nécessitent des paramètres pour s'exécuter, mais dans une macro ils sont toujours fixes. Le mécanisme des procédures d'implantation automatique (PIA) autorise la notion de paramètres formels. Les paramètres effectifs sont donnés lors de l'appel du PIA. L'introduction de paramètres renforce l'indépendance contextuelle des PIA vis à vis des macros, de plus il est désormais possible de créer des PIA plus généraux que les macros.

13.1 Construction d'un PIA

Deux moyens sont actuellement possibles pour construire des PIA. La première étape conseillée est la construction interactive du PIA. L'utilisateur utilise le mode d'apprentissage d'une macro ou d'un script pour une première ébauche. Cette manière de procéder est efficace car l'utilisateur voit directement le résultat d'exécution et peut ainsi corriger des éventuelles erreurs de construction. De plus, la méthodologie de conception utilisée pour l'ébauche est implicitement transcrite dans la macro ou le script. Un script ne pouvant en appeler un autre, nous préconisons l'utilisation des macros. Chaque macro nommée correspond alors à un PIA où le nom du PIA est le nom de la macro. Quand l'utilisateur est satisfait, il sauvegarde les macros sur fichier.

Un PIA est éditable. Dans le cas d'ébauche par macros, éditer le fichier macro correspondant. A chaque définition de PIA correspond un fichier. Si plusieurs macros ont été nécessaires pour la première ébauche, découpez-le en plusieurs fichiers (un par macro) ayant pour nom le nom de la macro (qui est aussi le nom du PIA).

L'utilisateur doit passer par une phase d'édition de texte pour apporter des modifications nécessaires à la bonne exécution du PIA. L'ébauche du PIA contient actuellement des caractères non éditables qui doivent être modifiés pour une meilleure lisibilité. Un PIA est par définition paramétrable, l'utilisateur doit introduire les paramètres désirés.

Un paramètre est noté par le caractère % suivi du numéro de paramètre. Ce paramètre peut être soit un paramètre d'une commande (nom d'un fichier par exemple), soit une commande elle-même (commande de déplacement). L'association paramètre effectif-formel se fait à l'appel du PIA, le premier paramètre effectif est associé au paramètre formel %1 et ainsi de suite. Le contrôle de type (caractère, entier, chaîne longue ou courte) est effectué en cours d'exécution du PIA. A l'appel d'un PIA, le système vérifie uniquement si tous les paramètres sont donnés. Bien entendu, un PIA peut appeler un autre PIA et lui passer comme paramètre un de ses propres paramètres. Attention cependant au numérotage : si PIA1 appelle PIA2 avec comme paramètre effectif %3, %3 devient %1 dans le corps de PIA2.

L'appel d'un PIA se fait par la commande "ESC p" suivi du nom de fichier correspondant au nom du PIA, puis la liste des paramètres effectifs. Si un PIA appelle un autre PIA construit comme une macro il est donc nécessaire de changer l'appel de cette macro en tenant compte des remarques ci-dessus.

Pour éviter d'éditer des textes contenant des caractères de contrôles, certains caractères doivent être mis en littéraux. Ainsi vous devez effectuer les modifications suivantes.

Le caractère ESC (qui est généralement donné par son code octal) doit être remplacé par la chaîne ESC.

Les caractères de contrôles ^a à ^z (donnés par leur code octal) doivent être remplacés par les chaînes ^A à ^Z.

Les espaces et les retour-chariots ne sont pas significatifs, il faut les remplacer par les chaînes SP et CR. L'utilisateur peut donc ajouter des espaces et des retour-chariots (véritables) dans le texte du PIA pour séparer les différentes commandes, mais bien entendu sans couper un paramètre en deux ou encore une commande.

Le concepteur peut passer directement à la phase d'édition du PIA. Il doit cependant suivre les directives précédentes pour taper les différentes commandes de son PIA. Ces PIA sont dans la version actuelle interprétés. Notons cependant la possibilité de contrôler l'exécution ou la non-exécution d'une commande par la commande "ESC u". En lui passant comme paramètre 1, la commande suivante est exécutée, en lui passant comme paramètre 0, la commande suivante n'est pas exécutée.

13.2 Exécution et exemple

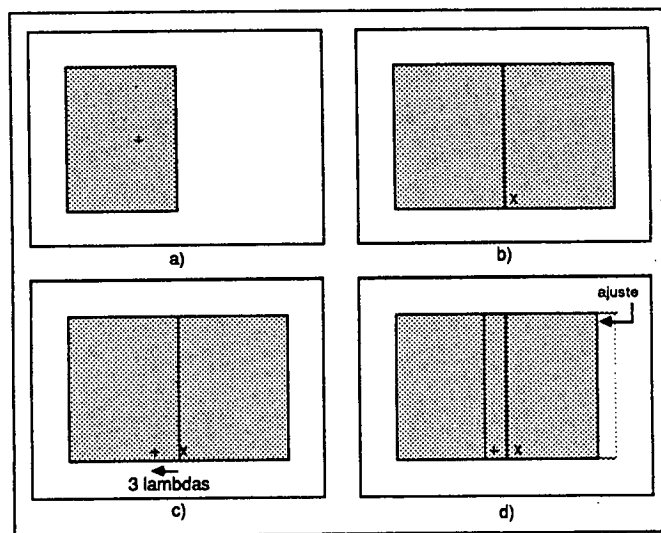
Prenons l'exemple de la réplication d'une cellule vers la droite. Pour généraliser cette fonction il faut prévoir le recouvrement des deux cellules (laissons de côté les éventuels problèmes d'orientation de cellules pour plus de simplicité). La résolution de ce problème peut se faire comme suit. Nous créons un PIA de nom repx et qui a comme paramètre formel le nombre de lambdas de recouvrements. Les commandes du PIA sont mises en littérales pour une meilleure compréhension du lecteur.

```
répliquer la cellule courante vers la droite.
marquer la position courante.
répéter la commande suivante %1 fois (%1 est le paramètre
formel).
se déplacer d'un lambda vers la gauche.
ajuster la cellule marquée.
```

L'appel de ce PIA se fait grâce à la commande ESC p repx 3, si 3 est le nombre de lambdas de recouvrement désiré. Une illustration de l'exécution de ce PIA est donné par la figure 10.

Lors de la présentation des macros, nous avons mis en évidence l'indépendance contextuelle pour l'exécution. Cette remarque est bien sur valable pour les PIA, et d'autant plus forte grâce au mécanisme de paramètre présenté. Il n'est pas toujours souhaitable d'utiliser les PIA à des niveaux de constructions de cellules. Les PIA nous semble par contre un outil indispensable de construction de structures régulières (ou semi-régulières) tel que mémoires ou PLAs à partir de cellules de base. Fréquemment, l'utilisateur construit un prototype pour une structure de quelques éléments. Le PIA en est dérivé simplement et paramétrisé ce qui permet alors des constructions de tailles variables.

Une autre utilisation fréquente consiste en la construction du PIA de routage des blocs du circuit, afin de mémoriser le placement des blocs. Des modifications simples du circuit (déplacement d'un bloc de 1 lambda vers la droite) deviennent lourdes à gérer avec un éditeur classique et obligent souvent l'utilisateur à refaire une grande partie de la connectique de son circuit. L'utilisation du PIA et des déplacements relatifs permet de reconstruire le circuit en un minimum de travail, il suffit généralement d'apporter quelques modifications au PIA. Signalons qu'un bon emploi du vérificateur de règles permet en même temps une vérification de cohérence du dessin reconstruit. Le vérificateur peut être appelé par un PIA et exécuté sur les bords des blocs afin de vérifier la cohérence des connexions.



Le PIA REPX réplique la cellule courante vers la droite, déplace la nouvelle cellule vers la gauche de n lambdas donné en paramètre. Cet exemple montre l'exécution de la commande exécute-pia REPX 3.

- a) Le curseur pointe sur une cellule
- b) Répliquer la cellule courante vers la droite
Mettre la marque courante
- c) Répéter la commande suivante %1 (3 fois dans ce cas)
Déplacer le curseur d'un lambda vers la gauche
- d) Ajuster la cellule marquée

Figure 11: Exemple d'exécution de PIA

14 Méthodologie de conception

La philosophie des commandes et des mécanismes présentés invite l'utilisateur à adopter une méthodologie de conception à la "Madmacs". Le système inclut de nombreuses facilités pour une utilisation des structures régulières préconisée par Mead et Conway. Le point fort de Madmacs permet de plus l'automatisation de la construction de structures semi-régulières. Madmacs invite donc les utilisateurs à ne pas foncer tête basse sur les problèmes mais à prendre du recul vis à vis du problème et regarder si une automatisation est possible.

CONCLUSION

Les commandes présentées ci-dessus donnent un aperçu de la puissance de cet éditeur graphique. Les premières utilisations de MADMACS pour la construction de circuits répondent parfaitement aux besoins des concepteurs : un outil entièrement interactif, souple d'utilisation (avec des commandes puissantes pour résoudre des problèmes complexes) et également extensible grâce aux macros et au PIA. L'utilisation de MADMACS vous permet de construire des circuits de plus en plus complexes.

MADMACS n'est dans sa version actuelle qu'un éditeur graphique. Il continue d'évoluer dans ce sens suite aux requêtes des concepteurs et aux nouvelles commandes intégrées. Il possèdera des interfaces vers des simulateurs électriques et logiques. Et surtout un mécanisme plus puissant de PIA: dans la version suivante les PIA pourront automatiquement être traduits en langage PASCAL et ainsi intégrer toute la puissance de ce langage.

REFERENCES

(Mead, 1980)

C. Mead, L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980.

(Guyot, 1982)

A. Guyot, A. Jerraya, J. Raymond, *Manuel d'utilisation du système LUCIE*, IMAG, Octobre 1982.

ANNEXE 1

Commandes de MADMACS par types

CONVENTIONS D'ECRITURE

^	Touche contrôle (CTRL) taper cette touche en simultanéité.
ESC	Touche escape (ESC) taper cette touche séquentiellement.
RC	Touche retour chariot.
SP	Touche espace.
T	Vecteur de translation T défini par la marque courante et la position actuelle du curseur.

COMMANDES SYSTEME

Environnement

q	Quitter MADMACS avec une validation par o (oui) ou par n (non).
^x^e	Envoyer une commande au système hôte, réponse dans la zone texte.
^xm	Envoyer une commande au système hôte, réponse dans la zone de dialogue.
ESC e	Appeler l'éditeur de textes du système hôte (EMACS sous MULTICS).
^g	Arrêter la commande en cours.
ESC u	Appliquer un facteur multiplicatif à la commande suivante.

Demande d'informations

?	Lister toutes les commandes (avec un peu d'informations) de MADMACS dans la zone texte.
p	Lister les noms des figures du buffer courant dans la zone de texte.
t	Lancer un script pour apprentissage de MADMACS (non implémenté).

Gestion des fichiers

^x^w	Ecrire le dessin sur le fichier de nom passé en paramètre. Vérification sur le nom du fichier et les accès.
^x^r	Lire un dessin sur le fichier de nom passé en paramètre. Vérification sur le nom du fichier et les accès.
^xi	Insérer un dessin, contenu dans un fichier, par rapport au curseur. Vérification sur le nom du fichier et les accès.

DEPLACEMENT RELATIF DU CURSEUR

Déplacement dans une cellule

D	Déplacer le curseur sur le côté inférieur de la cellule pointée.
L	Déplacer le curseur sur le côté gauche de la cellule pointée.
R	Déplacer le curseur sur le côté droit de la cellule pointée.
U	Déplacer le curseur sur le côté supérieur de la cellule pointée.
i	Si le curseur est sur le côté droit ou gauche, il va se positionner sur l'autre côté.
j	Si le curseur est sur le côté supérieur ou inférieur, il va se positionner sur l'autre côté.

Déplacement à une autre cellule

d	Déplacer le curseur à la cellule directement en dessous.
l	Déplacer le curseur à la cellule directement à gauche.
r	Déplacer le curseur à la cellule directement à droite.
u	Déplacer le curseur à la cellule directement au dessus.

Déplacement par rapport aux autres cellules

^d	Aligner le curseur par rapport à la première cellule de côté supérieur plus bas que la position actuelle du curseur.
^l	Aligner le curseur par rapport à la première cellule de côté droit à gauche de la position actuelle du curseur.
^r	Aligner le curseur par rapport à la première cellule de côté gauche à droite de la position actuelle du curseur.
^u	Aligner le curseur par rapport à la première cellule de côté inférieur au dessus de la position actuelle du curseur.

Déplacement par rapport aux connecteurs d'une figure

ESC 2	Déplacer le curseur au connecteur directement en dessous.
ESC 4	Déplacer le curseur au connecteur directement à gauche.
ESC 6	Déplacer le curseur au connecteur directement à droite.
ESC 8	Déplacer le curseur au connecteur directement au dessus.

Déplacement par rapport à la figure englobante

^zd	Déplacer le curseur sur le côté inférieur de la figure englobante.
^zl	Déplacer le curseur sur le côté gauche de la figure englobante.
^zr	Déplacer le curseur sur le côté droit de la figure englobante.

^zu Déplacer le curseur sur le côté supérieur de la figure englobante.

Déplacement par rapport à l'écran

^z^d Déplacer le curseur d'une demi-fenêtre vers le bas.
^z^l Déplacer le curseur d'une demi-fenêtre vers la gauche.
^z^r Déplacer le curseur d'une demi-fenêtre vers la droite.
^z^u Déplacer le curseur d'une demi-fenêtre vers le haut.

Remplacement du curseur

5 Replacer le curseur à sa véritable position. Utile sur terminal alphanumérique où la gestion du curseur est imparfaite.
; Replacer le curseur au point bas gauche de la figure englobante.
g Déplacer le curseur aux coordonnées absolues passées en paramètre.

DEPLACEMENT DU CURSEUR DANS LA GRILLE DE PAS LAMBDA

1 Déplacer le curseur d'un lambda vers le bas et d'un lambda vers la gauche.
2 Déplacer le curseur d'un lambda vers le bas.
3 Déplacer le curseur d'un lambda vers le bas et d'un lambda vers la droite.
4 Déplacer le curseur d'un lambda vers la gauche.
6 Déplacer le curseur d'un lambda vers la droite.
7 Déplacer le curseur d'un lambda vers le haut et d'un lambda vers la gauche.
8 Déplacer le curseur d'un lambda vers le haut.
9 Déplacer le curseur d'un lambda vers le haut et d'un lambda vers la droite

COMMANDES DE VISUALISATION

+ Effectuer un zoom de l'écran. Le curseur est remplacé au centre de l'écran.
- Effectuer un panoramique de l'écran. Le curseur est remplacé au centre de l'écran.
= Recentrer le curseur au milieu de l'écran avec déplacement du dessin.
\$ Rendre les connecteurs des figures visibles ou invisibles.
m Cartographier complètement la figure englobante (plan de masse).
Esc-1 Prendre comme facteur d'échelle un pixel = un lambda carré.
^p Repeindre l'écran.

P	Peindre l'écran.
*	Visualiser le dessin à plat sans tenir compte de la hiérarchie.
^z*	Visualiser la figure pointée à plat sans tenir compte de la hiérarchie.
^v	Repeindre l'écran avec uniquement les rectangles de couleur courante.

MANIPULATION DE LA HIERARCHIE

:	Descendre dans la hiérarchie.
,	Remonter dans la hiérarchie.

COMMANDES DE MANIPULATION DE CELLULES

Création de cellules

c	Créer un rectangle (définie par la marque et le curseur).
Esc-c	Choisir un autre crayon pour peindre un rectangle.
f	Créer une figure (définie par la marque et le curseur).
^xc	Créer une région (figure), toutes les cellules comprises dans cette région sont désormais définies dans la figure région.

Destruction de cellules

^c	Détruire le rectangle pointé.
^f	Détruire la figure pointée.
^k	Détruire le rectangle de couleur demandée en paramètre. En NMOS les couleurs sont d, p, m, i, c.
^a	Aplanir la hiérarchie, toutes les cellules définies dans la figure pointée sont remontées d'un niveau hiérarchique. La figure pointée est détruite.

Modification d'une cellule

^y	Ramener la dernière cellule détruite.
ESC w	Mémoriser la cellule pointée sans la détruire.
a	Ajuster la cellule. On lui applique une translation de vecteur T.
^e	Etendre le rectangle marqué en tenant compte du vecteur T.
s	Serrer le rectangle marqué en tenant compte du vecteur T.
M	Changer la couleur du rectangle pointé. En NMOS les couleurs sont d, p, m, i ou c.
^zc	Changer le nom de la figure pointée.
^zp	Afficher le nom de la figure pointée dans la zone

dialogue.
 n Nommer le rectangle pointé.
 N Changer le nom du rectangle pointé.
 ESC n Afficher le nom du rectangle pointé dans la zone de dialogue.
 ^n Détruire le nom du rectangle pointé.

Réplication de la cellule pointée

^x1 Répliquer la cellule pointée vers le bas et la gauche.
 ^x2 Répliquer la cellule pointée vers le bas.
 ^x3 Répliquer la cellule pointée vers le bas et la droite.
 ^x4 Répliquer la cellule pointée vers la gauche.
 ^x6 Répliquer la cellule pointée vers la droite.
 ^x7 Répliquer la cellule pointée vers le haut et la gauche.
 ^x8 Répliquer la cellule pointée vers le haut.
 ^x9 Répliquer la cellule pointée vers le haut et la droite.

Orientation d'une cellule

h Effectuer un miroir en x sur la cellule pointée.
 v Effectuer un miroir en y sur la cellule pointée.
 / Effectuer une rotation de 90 degrés sur la cellule pointée.
 \ Effectuer une rotation de -90 degrés sur la cellule pointée.

BUFFERS

^xb Changer de buffer, avec création s'il n'existe pas.
 ^x^b Lister tous les noms de buffers créés.
 ^x^k Vider le buffer courant.
 ^xk Détruire un buffer. Taper RC pour le buffer courant, puis le nom du nouveau buffer courant.
 ESC Forcer le flag du buffer courant à non modifié.

MARQUAGE

Mettre une marque

. Mettre la marque courante à la position du curseur.
 ^z. Mettre une marque nommée à la position du curseur.
 ^x. Marquer le rectangle de couleur courante.
 A Pointer l'autre cellule recouverte.

Déplacement à une marque

^x^x Echanger la position curseur avec la marque courante.
 ^zg Positionner le curseur sur la marque nommée.

^xh	Aligner la position du curseur avec la marque courante suivant l'axe des X.
^xv	Aligner la position du curseur avec la marque courante suivant l'axe des Y.
^x^h	Aligner la position du curseur avec la marque nommée suivant l'axe des X.
^x^v	Aligner la position du curseur avec la marque nommée suivant l'axe des Y.

SCRIPT

Création d'un script

ESC o	Ouvrir un nouveau script en apprentissage.
^o	Arrêter l'apprentissage du script en cours.

Exécution d'un script

ESC i	Exécuter un script.
--------------	---------------------

PIA

ESC p	Exécuter un pia (ne pas oublier de passer les paramètres effectifs lors de l'appel).
--------------	--

MACRO

Macro courante

^x(Passer en mode apprentissage Macro.
^x)	Sortir du mode apprentissage Macro.
^xe	Exécuter la macro courante.

Macro nommée

^xn	Donner un nom à la macro courante.
ESC x	Exécuter la macro de nom passé en paramètre.

Edition macro

^x*	Afficher la macro courante dans la zone texte.
ESC x show-macro	Afficher une macro nommée dans la zone texte.

Fichiers de macros

- ESC x load-macros Charger un fichier de macros nommées.
- ESC x save-macros Sauvegarder les macros nommées sur un fichier.

Outils de vérification

- ESC x vrd Appeler le vérificateur de règles de dessins sur la zone définie par la marque courante et le curseur.
- ESC x jtf Demander une justification des erreurs de dessins détectées par le vérificateur de règles de dessins.
- ESC x xtr Extraire la liste des transistors et noeuds électriques dans la zone définie par la marque courante et le curseur.

ANNEXE 2

Commandes MADMACS par ordre lexicographique

\$	connecteur-visible	connector-visibility
*	détail	detail
+	zoom	zoom-in
,	remonter-hiérarchie	go-up-hierarchy
-	panoramique	zoom-out
.	mettre-marque	set-the-mark
/	rotation(90)	rotate(90)
\	rotation(-90)	rotate(-90)
1	curseur-bas-gauche	cursor-down-left
2	curseur-bas	cursor-down
3	curseur-bas-droite	cursor-down-right
4	curseur-gauche	cursor-left
5	remplacer-curseur	fix-cursor-position
6	curseur-droite	cursor-right
7	curseur-haut-gauche	cursor-up-left
8	curseur-haut	cursor-up
9	curseur-haut-droite	cursor-up-right
:	descendre-hiérarchie	go-down-hierarchy
;	recadrer-curseur	cursor-tally
=	centrage-ecran	center-screen
?	lister-commandes	list-commands
a	ajuster-cellule	cell-adjust
c	créer-rectangle	rectangle-create
d	cellule-en-bas	cell-below
f	créer-figure	create-symbol
g	déplacer-curseur-en-absolu	move-the-cursor-to-absolute-coord.
h	miroirx	mirorx
i	autre-côté-cellule-axeX	switch-side-cell-along-Xaxis
j	autre-côté-cellule-axeY	switch-side-cell-along-Yaxis
l	cellule-à-gauche	cell-to-the-left
m	cartographier	floorplan
n	nommer-le-rectangle-pointé	name-the-pointed-rectangle
p	lister-figures	list-symbols
q	quitter	quit
r	cellule-à-droite	cell-to-the-right
s	serrer-rectangle	rectangle-compress
t	apprentissage	teach
u	cellule-au-dessus	cell-above
v	miroiry	miroiry
A	pointer-l'autre-cellule-recouverte	point-the-overlaped-cell
D	côté-bas-de-la-cellule	cell-bottom-side
L	côté-gauche-de-la-cellule	cell-left-side
M	change-couleur-rectangle	change-rectangle-color

N	change-le-nom-du-rectangle-pointé	change-the-pointed-rectangle-name
P	peindre-écran	paint-screen
R	côté-droit-de-la-cellule	cell-right-side
U	côté-haut-de-la-cellule	cell-top-side
^a	aplanir-hiérarchie	flatten-hierarchy
^c	détruire-rectangle	delete-rectangle
^d	aligner-cellule-en-dessous	align-cell-below
^e	étendre-rectangle	rectangle-stretch
^f	détruire-figure	delete-symbol
^g	arrêt-commande-en-cours	command-abort
^k	détruire-rectangle-colorié	delete-colored-rectangle
^l	aligner-cellule-à-gauche	align-cell-to-the-left
^n	détruire-le-nom-du-rectangle	delete-the-pointed-rectangle-name
^o	arrêt-apprentissage-script	stop-script-learn
^p	repeindre-écran	repaint-screen
^r	aligner-cellule-à-droite	align-cell-to-the-right
^u	aligner-cellule-au-dessus	align-cell-above
^v	peindre-avec-rectangle-couleur-crt	paint-with-current-colored-rectangle
^y	ramener	yank
^x*	éditer-macro	edit-macro
^x1	copier-cellule-vers-bas-gauche	copy-cell-left-down
^x2	copier-cellule-vers-bas	copy-cell-down
^x3	copier-cellule-vers-bas-droite	copy-cell-down-right
^x4	copier-cellule-vers-gauche	copy-cell-left
^x6	copier-cellule-vers-droite	copy-cell-right
^x7	copier-cellule-vers-haut-gauche	copy-cell-left-up
^x8	copier-cellule-vers-haut	copy-cell-up
^x9	copier-cellule-vers-haut-droite	copy-cell-up-right
^x(début-macro-aprentissage	begin-macro-learn
^x)	fin-macro-apprentissage	end-macro-learn
^x.	marquer-rectangle-de-couleur-crt	mark-the-current-color-rectangle
^xb	changer-buffer	select-buffer
^xc	créer-région	create-region
^xe	exécuter-macro-courante	exec-current-macro
^xh	aligne-position-marque-axeX	horizontal-alignment-on-mark
^xi	insérer-fichier	insert-file
^xk	détruire-buffer	delete-buffer
^xm	envoi-commande-système-hôte	comout-command
^xn	nommage-macro	name-current-macro
^xv	aligner-position-marque-axeY	vertical-alignment-on-mark
^x^b	lister-buffers	list-buffers
^x^e	envoi-commande-système-hôte	comout-command
^x^h	aligner-sur-marque-nommée-axeX	horizontal-alignment-on-named-mark
^x^k	vider-buffer	empty-buffer
^x^r	lire-fichier	read-file
^x^v	aligner-sur-marque-nommée-axeY	vertical-alignment-on-named-mark
^x^w	écrire-fichier	write-file
^x^x	échanger-curseur-avec-marque	exchange-cursor-with-mark
^z*	détailler-figure	detail-symbol
^z.	mettre-marque-nommée	set-named-mark
^zc	changer-nom-figure	change-figure-name
^zd	bas-dessin	layout-bottom
^zg	aller-marque-nommée	goto-named-mark
^zl	gauche-dessin	layout-left

^zp	afficher-nom-figure	print-symbol-name
^zr	droite-dessin	layout-right
^zu	haut-dessin	layout-top
^z^d	demi-fenêtre-bas	half-window-down
^z^l	demi-fenêtre-gauche	half-window-left
^z^r	demi-fenêtre-droite	half-window-right
^z^u	demi-fenêtre-haut	half-window-up
ESC ~	buffer-non-modifié	not-modified-buffer
ESC 1	échelle-1	scale-1
ESC 2	connecteur-en-bas	connect-below
ESC 4	connecteur-à-gauche	connect-to-the-left
ESC 6	connecteur-à-droite	connect-to-the-right
ESC 8	connecteur-au-dessus	connect-above
ESC c	choix-crayon	choose-pen
ESC e	appel-éditeur-de-texte (emacs)	call-text-editor
ESC i	exécution-script	exec-script
ESC n	liste-le-nom-du-rectangle-pointé	show-the-pointed-rectangle-name
ESC o	apprentissage-script	learn-script
ESC p	execute-pia	execute-alp
ESC u	multiplication	multiplier
ESC w	mémoriser-cellule-pointée	memorize-pointed-cell
ESC x	commande-étendue (macro)	extended-command (macro)

COMMANDES ETENDUES

ESC x load-macros	charge-fichier-macros	load-macrofile
ESC x save-macros	sauve-fichier-macros	save-macrofile
ESC x show-macro	éditer-macro-nommée	edit-named-macro
ESC x vrd	appel-vrd	call-drc
ESC x jtf	justifie-erreur-dessin	justify-rule-violation
ESC x xtr	appel-extracteur	call-extractor

ANNEXE 3

Erreurs détectées par le vérificateur de dessins

Cette annexe doit permettre à l'utilisateur une compréhension directe des erreurs de dessins détectées par le vérificateur de règles de dessins. Dans un premier temps, nous montrons les conventions des codes d'erreurs avec quelques exemples puis les cas particuliers.

Un code erreur est associé à une fenêtre. Plus particulièrement les erreurs de dessins sont rattachées au coin bas-gauche de la fenêtre. Par exemple le code erreur signale une faute d'écart diffusion vers le haut, cela signifie la présence de diffusion dans le coin bas-gauche et de la diffusion 2 lambdas au dessus.

Le type de l'erreur est codée sous forme d'une chaîne de caractères. La première lettre donne le type de l'erreur. La seconde lettre (et éventuellement la troisième) donne le nom de la couche technologique concernée par l'erreur. La dernière lettre indique une direction.

Types de l'erreur

L Largeur insuffisante.
E Espacement insuffisant.
d Débordement insuffisant.

Types des couches en technologie NMOS

D Diffusion	(couleur verte).
P Polysilicium	(couleur rouge).
M Métal	(couleur bleue).
I Implant	(couleur jaune).
C Contact	(couleur blanche).
G Glaçage	(couleur mauve).
T Transistor	(couleur verte et rouge).

Les directions

> Droite.
< Gauche.
^ Haut.
v Bas.
/ Diagonale.
\ Contre-diagonale.

Exemples commentés

ED>	Ecart insuffisant diffusion vers la droite.
dI^	Débordement insuffisant implant vers le haut.
LMv	Largeur insuffisante métal vers le bas.
EDP/	Ecart insuffisant diffusion polysilicium en diagonale.
dDT<	Débordement insuffisant diffusion par rapport au transistor vers la gauche.

Cas particuliers

La plupart des cas particuliers sont liés aux ponts métalliques et aux contacts. Ils sont présentés ci-après.

CT	Une erreur de type CT (2 premiers caractères) signifie une erreur d'écart entre le contact et le transistor.
TC	Une erreur de type TC (2 premiers caractères) signifie une erreur d'écart entre le transistor et le contact.
dMC	Une erreur de type d sans direction donnée signifie un débordement insuffisant par rapport au contact. (autre dDPC : débordement insuffisant de diffusion ou de polysilicium par rapport au contact)

ANNEXE 4

Différentes constantes du système

Dans cette annexe, une liste non exhaustive des constantes du système est présentée. Il s'agit dans la plupart des cas des bornes supérieures des tables gérées par Madmacs. Les valeurs données sont celles de la version implantée sur une machine SM90 avec le système d'exploitation SMX. La configuration mémoire est de 2,5 MO. Cette machine ne possède pas de gestion de mémoire virtuelle et impose de charger en mémoire tout le code et les données. Sur d'autres machines les constantes données dans cette annexe sont plus élevées.

1 Taille des données

Nombre de définition de figures < 500.

Taille globale du code <= 20000 enregistrements.

Taille du code d'une figure <= 2500 enregistrements.

Profondeur dans la hiérarchie <= 20.

Dans le cas d'un débordement, Madmacs envoie un message d'erreur à l'utilisateur et refuse toute nouvelle création.

2 Constantes systèmes

Nombre de marques nommées <= 10

Nombre de buffers <= 2.

Profondeur d'appel <= 10.

Taille d'une macro <= 300 caractères.

Nombre de macros nommées <= 100.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique